DESARROLLO DE SOFTWARE DE DINÁMICA DE VUELO DE UN ORBITADOR LUNAR

Álvaro Morejón Barberá

INDICE

1. SUMARIO

El 4 de Octubre de 1957 la unión soviética ponía en órbita el primer satélite artificial alrededor de la tierra, el Sputnik I. Dos años después con el programa Luna 1, la Unión Soviética demostró una vez más su capacidad tecnológica al enviar la primera sonda no tripulada a la Luna. No obstante, a pesar de haber conseguido liderar la carrera espacial en la mayoría de los hitos alcanzados hasta el momento, el 21 de Julio de 1969, Neil Amstrong se convirtió ante los ojos de millones de personas, en el primer hombre en dejar su huella sobre la Luna.

A pesar de haber tenido que esperar hasta la segunda mitad del siglo XX para concluir éste hito, el sueño de alcanzar la Luna data de una época muy anterior a los ordenadores y nunca habría sido posible sin la contribución escrita en tinta y papel de la mecánica celeste de Sir Isaac Newton y los estudios posteriores de Johannes Kepler.

El objeto de éste trabajo pretende crear un código que nos permita calcular la trayectoria descrita así como las condiciones de lanzamiento óptimas de una nave con la misión de viajar hasta la luna situarse en una órbita estable alrededor de ésta.

Nuestro objetivo es inherente a un sistema gravitatorio tremendamente complejo como es el sistema Tierra-Luna, donde el efecto gravitatorio del sol o de otros planetas, la radiación solar, las irregularidades de la simetría de los astros, la variación de masa de la nave por el gasto de consumibles y otros cientos de factores, pueden aumentar su complejidad hasta el grado que se considere conveniente. Por tanto, un objetivo añadido al desarrollo código es una descentralización que permite modificar módulos independientes para aumentar la complejidad del sistema hasta el punto que se considere necesario.

Para cumplir nuestro propósito, partiremos de la solución analítica de un modelo físico simplificado basado en una serie de hipótesis que detallaremos más adelante con el objetivo de obtener una primera aproximación del lanzamiento. A partir de ésta arrancaremos un proceso iterativo donde se realizarán varias llamadas a un módulo de cálculo externo basado en un modelo físico mucho más complejo que nos permita situar con mayor exactitud nuestra nave y calcular el incremento necesario para corregir la trayectoria hasta describir la órbita alrededor de la Luna. Dado que éste proceso puede realizarse de infinitas formas, se ha decidido incluir en el código la programación de una matriz de lanzamientos que nos permita hacer una comparación del gasto energético con el objetivo de minimizar el coste de la misión.

El programa se ha diseñado con una estructura modular para permitr su ampliación. La parte inicial está estructurada en funciones de propósitos independientes para permitir su modificación o adaptación de forma sencilla y el motor fundamental del cálculo de trayectorias, está basado en un programa externo con el que se interactúa a través de ficheros permitiendo así su actualización por una versión más potente. Las salidas de datos del código son escritas en ficheros de resultados, que pueden servir como datos de partida para otros módulos de cálculo.

En definitiva, el objeto del proyecto, pretende encontrar una solución numérica basada en las ecuaciones analíticas de un modelo simplificado, que nos acerque al complejo problema del lanzamiento lunar y estableciendo las bases de un sistema de cálculo que pueda ampliarse eliminando todas las simplificaciones hasta convertirse un problema tan complejo como queramos.

A pesar de su reducida complejidad inicial, no deja de ser una introducción a los conceptos físicos que han servido de base para permitir al hombre ser capaz de alcanzar la Luna.

2. BIBLIOGRAFÍA Y SOFTWARE UTILIZADO

BIBLIOGRAFIA

- *FUNDAMENTALS OF ASTRODYNAMICS* (Roger R. Bate, Donald D. Mueller, Jerry E. White) DOVER PUBLICATIONS ,INC. New York 1971.
- *GUÍA DEL FIRMAMENTO*. "Sexta edición". Comellas, José Luis. (1996). Editorial Rialp.
- *GLOSARIO SELENOGRÁFICO*, José Carlos Violat Bordonau. España, 2006.

SOFTWARE

- DEV C++. GNU GENERAL PUBLIC LICENSE, Version 2, June 1991, Copyright (C) 1989, 1991 Free Software Foundation, Inc. 675 Mass Ave, Cambridge, MA 02139, USA.
- GNU OCTAVE. version 3.2.2.
- Ms Office Excel 2007.

ENLACES

- http://es.wikipedia.org/wiki/Kepler
- http://es.wikipedia.org/wiki/Tycho_Brahe

3. INTRODUCCIÓN

Sin mayor interés que establecer conceptos básicos que permitan al lector seguir con la lectura del proyecto sin tener que referirse constantemente a un libro, vamos a exponer una breve introducción a los conceptos mecánicos que han servido de base para el desarrollo de éste texto. No obstante, quisiera animar al lector a que amplíe dichos conceptos profundizando en la lectura de alguno de los libros detallados en la bibliografía.

3.1 LOS PRIMEROS PASOS

Antes de comenzar con la exposición de conceptos físicos y matemáticos, considero importante el detenerse un instante a descubrir los acontecimientos que llevaron al estudio y descubrimiento de la mecánica celeste.

En 1642, dos grandes físicos de mentalidades completamente opuestas comenzaron a estudiar el movimiento de los planetas alrededor del sol, con la intención de hallar una representación matemática que permitiese describir el movimiento celeste de los planetas, Tycho Brahe y Johannes Kepler.

Aunque en realidad Tycho Brahe era únicamente un astrónomo, sus precisas anotaciones sobre la posición de los planetas, fueron la base que permitió a Johannes Kepler establecer una serie de leyes bajo las cuales se regía el movimiento de los planetas que posteriormente, servirían de base fundamental para el desarrollo de la ley gravitacional de Newton.

Hasta el momento, el movimiento de los planetas se consideraba según la hipótesis de Aristóteles como un movimiento circular alrededor del sol (Sistema Heliocéntrico) por lo que las mediciones realizadas, no coincidían con las posiciones teóricas explicadas a partir de la hipótesis de Aristóteles.

TYCHO BRAHE

Tycho Brahe fue el último de los grandes astrónomos observadores de la era previa a la invención del telescopio.

El 24 de agosto de 1563, mientras estudiaba en Leipzig, ocurrió una conjunción de Júpiter y Saturno, suceso predicho por las tablas astronómicas existentes.

Sin embargo, Tycho se dio cuenta de que todas las predicciones sobre la fecha de la conjunción estaban equivocadas en días o incluso meses. Este hecho tuvo una gran influencia sobre él. Brahe se percató de la necesidad de compilar nuevas y precisas observaciones planetarias que le permitieran realizar tablas más exactas.

Durante su carrera científica persiguió con ahínco este objetivo. Así desarrolló nuevos instrumentos astronómicos. Con ellos fue capaz de realizar un preciso catálogo estelar de más de 1000 estrellas (777 de ellas con una precisión muy elevada) cuyas posiciones estaban medidas con una precisión muy superior a la alcanzada hasta entonces. Las mejores medidas de Tycho alcanzaban precisiones de medio minuto de arco.

Estas medidas le permitieron mostrar que los cometas no eran fenómenos meteorológicos sino objetos más allá de la Tierra. Sus instrumentos científicos fueron ampliamente copiados en Europa. Tycho fue el primer astrónomo en percibir la refracción de la luz, elaborar una completa tabla y corregir sus medidas astronómicas de este efecto.

El conjunto completo de observaciones de la trayectoria de los planetas fue heredado por Johannes Kepler, ayudante de Brahe en aquel tiempo. Gracias a estas detalladas observaciones Kepler sería capaz, unos años más tarde, de encontrar las hoy denominadas leyes de Kepler que gobiernan el movimiento planetario.

A pesar de que a partir de las leyes de Kepler se podía describir de forma matemática el movimiento de los planetas, éste no había sido capaz de encontrar una explicación física a dicho movimiento.

En 1665 Newton desarrolló un estudio donde describía las leyes gravitacionales, sin embargo, éste no sería publicado hasta 20 años más tarde. Una apuesta entre Edmund Halley, Christofer Wren y Robet Hooke fue la chispa necesaria para desempolvar dichos estudios a partir de una pregunta casi retórica, plantada por Halley a Newton acerca de la trayectoria que describirían los planetas si fuesen empujados por el sol con una fuerza inversamente proporcional a su distancia. La respuesta dada por el genio sin ningún atisbo de duda desempolvó los estudios que había comenzado veinte años antes. “Una elipse, por supuesto”. Dos años más tarde, se publicaría el “Principia Mathematica” y con él los secretos del movimiento planetario.

3.1.1 LEYES DE KEPLER

Antes de que se desarrollara la ley gravitacional, Kepler, extrajo de las observaciones de Tycho Brahe una serie de leyes que condicionaban el movimiento de los planetas, que sirvieron de base para la explicación física desarrollada por Newton posteriormente.

- Los planetas tienen movimientos elípticos alrededor del Sol, estando éste situado en uno de los focos de la elipse.
- Los planetas, en su recorrido por la elipse, barren áreas iguales en el mismo tiempo.
- El cuadrado de los períodos de los planetas es proporcional al cubo de la distancia media al Sol.

3.1.2 LEYES DE NEWTON

Las Leyes de Newton, también conocidas como *Leyes del movimiento de* Newton, son tres principios a partir de los cuales se explican la mayor parte de los problemas planteados por la dinámica, en particular aquellos relativos al movimiento de los cuerpos. Aunque incluyen ciertas definiciones y en cierto sentido pueden verse como axiomas, Newton afirmó que estaban basadas en observaciones y experimentos cuantitativos; ciertamente no pueden derivarse a partir de otras relaciones más básicas. La demostración de su validez radica en sus predicciones... La validez de esas predicciones fue verificada en todos y cada uno de los casos durante más de dos siglos.

Constituyen, junto con la transformación de Galileo, la base de la mecánica clásica. Al combinar estas leyes con la Ley de la gravitación universal, se pueden deducir y explicar las Leyes de Kepler sobre el movimiento planetario.

En base a estas leyes, podemos explicar el movimiento de los astros y de los proyectiles artificiales, así como toda la mecánica de funcionamiento de las máquinas.

La dinámica de Newton, también llamada dinámica clásica, sólo se cumple en los sistemas de referencia inerciales; es decir, sólo es aplicable a cuerpos cuya velocidad dista considerablemente de la velocidad de la luz (que no sobrepasen los 300,000 km/s); la razón estriba en que cuanto más cerca esté un cuerpo de alcanzar esa velocidad (lo que ocurriría en los sistemas de referencia no-inerciales), más posibilidades hay de que incidan sobre el mismo una serie de fenómenos denominados efectos relativistas o fuerzas ficticias, que añaden términos suplementarios capaces de explicar el movimiento de un sistema cerrado de partículas clásicas que interactúan entre sí. El estudio de estos efectos (aumento de la masa y contracción de la longitud, fundamentalmente) corresponde a la teoría de la relatividad especial, enunciada por Albert Einstein en 1905.

- **Principio de inercia** Todo cuerpo persevera en su estado de reposo o movimiento uniforme y rectilíneo a no ser en tanto que sea obligado por fuerzas impresas a cambiar su estado.

- **Ley de Fuerza.** El cambio de movimiento es proporcional a la fuerza motriz impresa y ocurre según la línea recta a lo largo de la cual aquella fuerza se imprime.

- **Ley de acción y reacción.** Con toda acción ocurre siempre una reacción igual y contraria: o sea, las acciones mutuas de dos cuerpos siempre son iguales y dirigidas en direcciones opuestas.

JOHANNES KEPLER

Después de estudiar teología en la universidad de Tubinga, incluyendo astronomía con un seguidor de Copérnico, enseñó en el seminario protestante de Graz. Kepler intentó comprender las leyes del movimiento planetario durante la mayor parte de su vida. En un principio Kepler consideró que el movimiento de los planetas debía cumplir las leyes pitagóricas de la armonía. Esta teoría es conocida como la música o la armonía de las esferas celestes. En su visión cosmológica no era casualidad que el número de planetas conocidos en su época fuera uno más que el número de poliedros perfectos. Siendo un firme partidario del modelo copernicano, intentó demostrar que las distancias de los planetas al Sol venían dadas por esferas en el interior de poliedros perfectos, anidadas sucesivamente unas en el interior de otras. En la esfera interior estaba Mercurio mientras que los otros cinco planetas (Venus, Tierra, Marte, Júpiter y Saturno) estarían situados en el interior de los cinco sólidos platónicos correspondientes también a los cinco elementos clásicos.

En 1596 Kepler escribió un libro en el que exponía sus ideas. *Misterium Cosmographicum* (*El misterio cósmico*). Siendo un hombre de gran vocación religiosa, Kepler veía en su modelo cosmológico una celebración de la existencia, sabiduría y elegancia de Dios. Escribió: «yo deseaba ser teólogo; pero ahora me doy cuenta a través de mi esfuerzo de que Dios puede ser celebrado también por la astronomía».

En 1600 acepta la propuesta de colaboración del astrónomo imperial Tycho Brahe, que a la sazón había montado el mejor centro de observación astronómica de esa época. Tycho Brahe disponía de los que entonces eran los mejores datos de observaciones planetarias pero la relación entre ambos fue compleja y marcada por la desconfianza. No será hasta 1602, a la muerte de Tycho, cuando Kepler consiga el acceso a todos los datos recopilados por Tycho, mucho más precisos que los manejados por Copérnico. A la vista de los datos, especialmente los relativos al movimiento retrógrado de Marte se dio cuenta de que el movimiento de los planetas no podía ser explicado por su modelo de poliedros perfectos y armonía de esferas. Kepler, hombre profundamente religioso, incapaz de aceptar que Dios no hubiera dispuesto que los planetas describieran figuras geométricas simples, se dedicó con tesón ilimitado a probar con toda suerte de combinaciones de círculos. Cuando se convenció de la imposibilidad de lograrlo con círculos, usó óvalos. Al fracasar también con ellos, «sólo me quedó una carreta de estiércol» y empleó elipses. Con ellas desentrañó sus famosas tres leyes (publicadas en 1609 en su obra *Astronomia Nova*) que describen el movimiento de los planetas. Leyes que asombraron al mundo, le revelaron como el mejor astrónomo de su época, aunque él no dejó de vivir como un cierto fracaso de su primigenia intuición de simplicidad (¿por qué elipses, habiendo círculos?). Sin embargo, tres siglos después, su intuición se vio confirmada cuando Einstein mostró en su Teoría de la Relatividad general que en la geometría tetradimensional del espacio-tiempo los cuerpos celestes siguen líneas rectas. Y es que aún había una figura más simple que el círculo: la recta.

En 1627 publicó las *Tabulae Rudolphine*, a las que dedicó un enorme esfuerzo, y que durante más de un siglo se usaron en todo el mundo para calcular las posiciones de los planetas y las estrellas. Utilizando las leyes del movimiento planetario fue capaz de predecir satisfactoriamente el tránsito de Venus del año 1631 con lo que su teoría quedó confirmada.

Escribió un biógrafo de la época con admiración, lo grande y magnífica que fue la obra de Kepler, pero al final se lamentaba de que un hombre de su sabiduría, en la última etapa de su vida, tuviese demencia senil, llegando incluso a afirmar que "las mareas venían motivadas por una atracción que la luna ejercía sobre los mares...". un hecho que fue demostrado años después de su muerte.

4. OBJETO DEL PROYECTO

El objeto de éste trabajo pretende crear un código que sirva de esqueleto para un programa de cálculo de trayectorias de lanzamientos lunares. El programa debe ser lo suficientemente flexible como para poder ampliar su complejidad hasta el punto que se considere necesario.

En primera instancia, el código debe leer las condiciones iniciales que condicionarán el lanzamiento y calcular los datos necesarios para arrancar el modelo de cálculo más complejo. Ésta primera parte, debe estar estructurada en funciones que permitan su adaptación para aumentar, o simplemente modificar, los condicionantes del lanzamiento, o para aumentar la complejidad del cálculo analítico que nos proporciona la primera aproximación con la intención de acelerar el proceso iterativo y debe tomar de datos de una fuente externa para poder ser lanzado desde otros programas.

En segundo lugar, el programa debe generar un proceso iterativo de cálculo que a través de un módulo completamente externo, obtenga la trayectoria óptima para cumplir la misión. Dado que el modelo de cálculo complejo, está basado un programa independiente, puede ser sustituido por versiones mejoradas que aumenten la precisión de las estimaciones.

Por último, el programa debe escribir los resultados en ficheros externos, que puedan ser utilizados por otros programas para permitir la integración completa del programa en un proyecto de mayor envergadura.

En definitiva, se trata de crear una base simplificada que permita el desarrollo de un programa de cálculo que pueda llegar a ampliarse lo suficiente como para ser integrado como parte de los sistemas de navegación de una nave espacial.

5. FUNDAMENTOS DE ASTRODINÁMICA

En éste apartado, trataremos de resumir la base física y matemática que nos permiten describir el comportamiento de la aeronave en el lanzamiento. La intención de los párrafos siguientes no va más allá de transmitir el concepto físico sobre el que se basa toda la metodología de cálculo que permita al lector entender mejor la estructura del algoritmo, por lo que insistiré en la explicación de los conceptos sin profundizar demasiado.

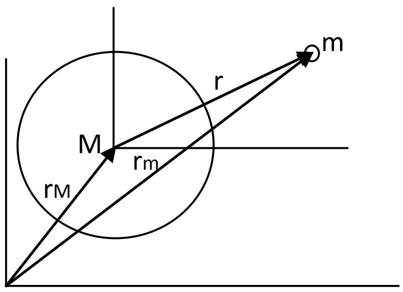
Si tratamos de hacer un planteamiento del problema que nos ocupa. Encontraremos que esencialmente no es más que un sistema de tres cuerpos de masas muy diferentes, sometido a una serie de esfuerzos. Aunque el problema real puede ser tremendamente complicado si tenemos en cuenta la interacción del resto de cuerpos celestes, las geometrías reales de los cuerpos, las variaciones de masa debidas al consumo de combustible y otros muchos factores que influyen en el desarrollo de la trayectoria, podemos establecer un punto de partida en un sistema simplificado, aislado y compuesto por cuerpos de geometría esférica que posteriormente podremos complicar teniendo en cuenta cada vez más agentes externos.

Como punto de partida, trataremos de explicar el comportamiento de dos cuerpos interactuando entre sí, situación más sencilla de la ley gravitacional.

5.1 MECANISMO DE ORBITACIÓN ENTRE DOS CUERPOS

En primer lugar, vamos a mostrar un breve desarrollo de la interacción que se produce entre dos cuerpos de masa muy distinta, y en base a las leyes de Newton extraeremos las ecuaciones que servirán como punto de partida para definir la trayectoria descrita por nuestra nave.

Imaginemos dos cuerpos, de masas distintas y geometría perfectamente esférica, situados a una distancia D el uno del otro referenciados sobre un sistema completamente aislado de la actuación gravitatoria de cualquier otro cuerpo.



Si aplicamos la ley de Newton, encontramos que los cuerpos se atraerán con una fuerza proporcional a sus masas e inversamente proporcional al cuadrado de su distancia en la dirección definida por la unión de sus centros $\vec{r}$. Si aplicamos la segunda ley de Newton, podemos definir ésta fuerza como,

$$m\ddot{\vec{r}}_m=-\frac{GMm}{r^2}\cdot\frac{\vec{r}}{r}$$

$$M\ddot{\vec{r}}_M=\frac{GMm}{r^2}\cdot\frac{\vec{r}}{r}$$

Donde G es la constante gravitatoria universal. Referenciando las ecuaciones a un sistema situado en el centro del mayor de ellos y aplicando el balance de equilibrio de fuerzas (2º ley de Newton), llegamos a,

$$\ddot{\vec{r}}=-\frac{G(M+m)}{r^3}.\vec{r}$$

Si la masa del segundo cuerpo es despreciable respecto al primero, como sucede en el caso de un satélite artificial, respecto a un planeta, podemos asumir que no tendrá apenas influencia y despreciarla. Quedando la ecuación de la siguiente forma,

$$\ddot{\vec{r}}=-\frac{GM}{r^3}.\vec{r}$$

Llegando a,

$$\boxed{\ddot{\vec{r}}+\frac{\mu}{r^3}.\vec{r}=0}$$

Ec. 1

$G = 6'67.10^{-8}$ dina.cm^2/gm^2

Es conveniente definir el parámetro gravitacional μ ya que es un valor característico de cada planeta.

5.1.2 CONSTANTES DE MOVIMIENTO

Aplicando las tres leyes de Newton sobre nuestro sistema, obtendremos las ecuaciones básicas que nos permitirán definir las trayectorias de los lanzamientos, así como algunas constantes características de las que conviene explicar su sentido físico.

5.1.2.1 Ley de conservación de la energía

Aplicando la ley de la conservación de la energía (3º ley de Newton) llegamos a la siguiente ecuación,

$$\frac{d}{dt}\left(\frac{v^2}{2}+\left(c-\frac{\mu}{r}\right)\right)=0$$

$$\varepsilon=\frac{v^2}{2}+\left(c-\frac{\mu}{r}\right)$$

Dado que la derivada del balance es igual a cero, su integral debe ser constante. Esto quiere decir que el balance energético se mantiene constante en todo momento. Definimos ésta constante como **Energía Mecánica Específica (ε).**

El primer término de la energía, corresponde a la energía cinética y el segundo a la potencial.

La constante c depende del punto que tomemos como origen de la energía potencial, o lo que es lo mismo, a qué distancia del origen queremos definir el valor nulo de la energía potencial. Normalmente, tomaremos el origen de referencia en el infinito, lo que implica que siempre tendremos una energía potencial negativa, resultando la ecuación anterior como:

$$\boxed{\varepsilon=\frac{v^2}{2}\pm\frac{\mu}{r}}$$

Ec. 2

5.1.2.2 Ley de conservación del momento angular

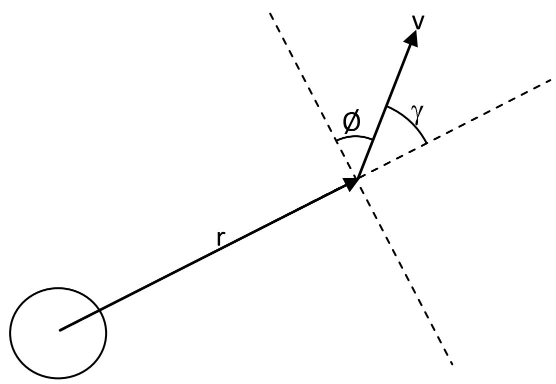
Aplicando la segunda ley de Newton, llegamos a,

$$\frac{d}{dt}(\vec{r}\times\vec{v})=0$$

$$\vec{h}=\vec{r}\times\vec{v}$$

Por tanto, el producto vectorial entre el vector posición y el vector velocidad, es otro vector constante perpendicular al plano formado por los anteriores, que denominaremos Momento Angular Específico.

Si definimos un eje local situado en la posición actual y formado por la dirección del vector posición y su normal en el plano definido por r y v, podemos definir el vector momento a partir de sus ángulos característicos.



Llamamos **Angulo de apogeo (Zenith Angle)** a γ y **ángulo de trayectoria de vuelo (fly path angle)** a **Ø.** Éste último es el ángulo que más utilizaremos.

Utilizando éstos ángulos, podemos expresar el módulo del momento angular a partir de ellos.

$$\boxed{h=r.v.\cos\emptyset}$$

Ec. 3

$$h=r.v.\text{sen}\gamma$$

5.1.3 ECUACIÓN DE LA TRAYECTORIA

Para poder definir la trayectoria, debemos obtener una ecuación que defina a partir de las constantes anteriores (Momento y energía) la evolución de vector radio. Para esto, integraremos la ecuación del movimiento [1] para obtener la siguiente expresión.

$$r = \frac{h^2/\mu}{1 + (B/\mu)cos\upsilon}$$

Donde υ es un ángulo característico formado por el vector B y el vector radio.

Si comparamos la ecuación anterior con la ecuación característica general de una sección cónica, podemos comprobar que son formalmente iguales.

$$\boxed{r = \frac{p}{1+e\cos\upsilon}}$$

Ec. 4

De ésta similitud, y las conclusiones anteriores, podemos resumir lo siguiente,

1- La única trayectoria posible que puede describir el movimiento de dos cuerpos en atracción gravitacional, pertenece a la familia de secciones cónicas.
2- El cuerpo gravitacional principal (de mayor masa) debe estar situado en el foco de la órbita cónica.
3- La energía mecánica no puede variar a lo largo de ésta órbita.
4- El momento angular no puede variar a lo largo de ésta órbita.
5- El movimiento orbital tiene lugar en un espacio fijo en el espacio inercial.

Podemos definir las siguientes relaciones, que caracterizaran el aspecto de la trayectoria que describirá el cuerpo de menor masa, alrededor del foco.

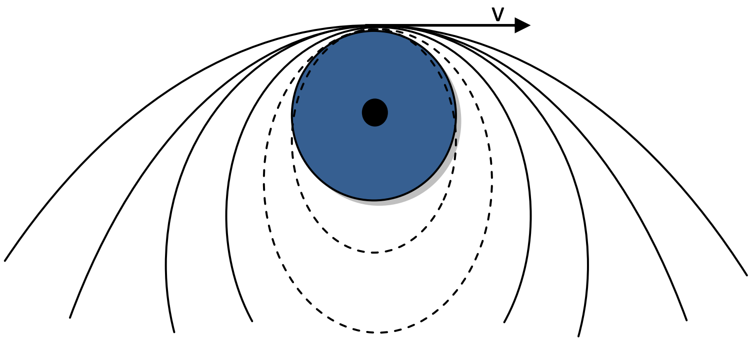
$$\boxed{p = \frac{h^2}{\mu}}$$

Ec. 6

$$\boxed{e = \sqrt{1 + \frac{2\varepsilon h^2}{\mu^2}}}$$

Ec. 5

Analizando éstas relaciones desde un punto de vista físico, y teniendo en cuenta el significado geométrico de éstos parámetros[1] , podemos observar como el parámetro p, crece exponencialmente en función del valor del momento angular *h*. Esta relación puede verse representada en la siguiente figura de forma mucho más intuitiva.



En función de la velocidad que le imprimamos al proyectil, la cónica descrita pos su trayectoria será mayor y por tanto su parámetro más ancho. Dado que la velocidad es tangente a la trayectoria en todos los puntos, el ángulo de vuelo Ø es igual a 0 en los extremos de la trayectoria, el perigeo y el apogeo. De ésta conclusión y teniendo en cuenta [3] podemos extraer la siguiente relación en los puntos extremos.

$$h = r_p v_p = r_a v_a$$

Sustituyendo éste valor en la ecuación de la energía y aplicando las relaciones de los cuerpos cónicos descritas en el Anexo II, llegamos a la siguiente ecuación.

$$\boxed{\boldsymbol{\varepsilon = -\frac{\mu}{2a}}}$$

Ec. 7

Dado que la excentricidad e, puede definirse a partir de la directriz y el parámetro, somos capaces de representar de definir la curva cónica que describe la trayectoria de nuestro cuerpo en función de su energía y su momento, o lo que es lo mismo, de su velocidad y su posición.

[1] Ver Anexo II

5.1.4 TRAYECTORIAS CARACTERÍSTICAS

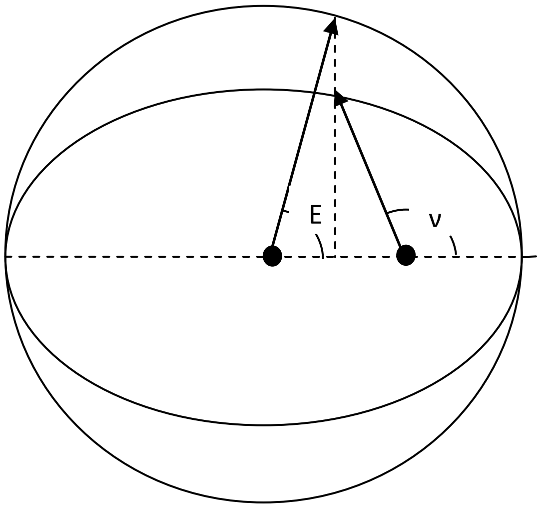
Para tener una primera ubicación de las posibles trayectorias cónicas, a continuación presentaremos una tabla con los parámetros característicos y el caso más habitual de cada una de las curvas cónicas.

TRAYECTORIA	EJEMPLO	PARAMETROS CARACTERÍSTICOS
ORBITA ELIPTICA	La mayoría de los planetas realizan un movimiento periódico describiendo una órbita elíptica. Denominamos periodo, al tiempo que tarda un satélite en describir una órbita completa alrededor del foco.	**Periodo**. $\Gamma = \frac{2\pi}{\sqrt{\mu}} a^{3/2}$
ORBITA CIRCULAR	En realidad es un caso especial de la elipse en la que los semiejes son iguales. Es la órbita más sencilla que puede describir un satélite. En éste caso, la velocidad descrita por el cuerpo es constante (V_{CS})	**Periodo,** $\Gamma = \frac{2\pi}{\sqrt{\mu}} {r_{CS}}^{3/2}$ **Velocidad satélite,** $V_{CS} = \sqrt{\frac{\mu}{r_{CS}}}$
ORBITA PARABÓLICA	En la órbita parabólica, el semieje mayor es infinito. Un satélite siguiendo ésta trayectoria, dejaría de orbitar alrededor del foco para perderse en el infinito. Es por tanto, la trayectoria que debe describir una aeronave para escapar del campo gravitatorio de un planeta. Definimos la velocidad de escape como la velocidad mínima que debe tener una aeronave para describir una trayectoria parabólica y por tanto escapar del campo gravitatorio de un planeta.	**Velocidad de escape,** $v_{esc} = \sqrt{\frac{2\mu}{r}}$
ORBITA HIPERBÓLICA	La hipérbola, es la cónica que debería describir un cuerpo, si quisiéramos tener velocidad extra una vez hemos escapado del campo gravitatorio. Es la órbita que se suele describir en los vuelos tripulados, donde el tiempo de vuelo es una factor que puede primar sobre el gasto energético debido a las reservas de oxígeno y comida de la nave.	**Velocidad de exceso,** $\boldsymbol{v_\infty^2 = v_0^2 - v_{esc}^2}$

5.2 POSICIÓN Y VELOCIDAD COMO FUNCIÓN DEL TIEMPO

Por el momento, somos capaces de describir la trayectoria de un satélite lanzado desde la luna a partir de las condiciones iniciales del lanzamiento. Sin embrago, cuando nos planteamos un lanzamiento con una trayectoria de destino, como es el envío de un satélite a la Luna, es imprescindible poder estimar el tiempo que durará nuestro vuelo ya que al ser la luna un cuerpo también en movimiento, determinará dónde podremos encontrarla una vez finalizada la trayectoria.

Para determinar el tiempo de vuelo, Kepler utilizó un sistema basado en la relación existente entre el área descrita por trayectoria elíptica como función del ángulo **v** y la descrita por la circunferencia que circunscribe a dicha elipse, en función del ángulo **E** (Eccentric anomally).



Relacionando éstas áreas y el periodo de rotación de ambas curvas llegamos a la siguiente expresión.

$$t - T = \sqrt{\frac{a^3}{\mu}}(E - \mathrm{sen}E)$$

Al término n, se le conoce como “Mean motion” y permite definir la expresión anterior como una definición de M (Mean Anomally) conocida como “*La ecuación de Kepler”.*

$$n = \sqrt{\frac{\mu}{a^3}}$$

$$M = n(t - T) = E - e.\mathrm{sen}E$$

La relación entre los ángulos ν y E mencionada anteriormente, se resume en la siguiente ecuación.

$$\cos E = \frac{e + \cos\nu}{1 + e.\cos\nu}$$

En general, para una trayectoria desde un tiempo t_0 hasta t, puede escribirse como,

$$\boxed{t - t_0 = \sqrt{\frac{a^3}{\mu}}\left[2k\pi + (E - e.\,\mathrm{sen}E) - (E_0 - e.\,\mathrm{sen}E_0)\right]}$$

Ec. 8

Siguiendo un procedimiento parecido, podemos llegar a las siguientes ecuaciones para una trayectoria parabólica e hiperbólica.

TRAYECTORIA PARABÓLICA

$$t - t_0 = \frac{1}{2\sqrt{\mu}}\left[\left(pD + \frac{1}{3}D^3\right) - \left(pD_0 + \frac{1}{3}{D_0}^3\right)\right]$$

$$D = \sqrt{p}.\tan\frac{\nu}{2}$$

TRAYECTORIA HIPERBÓLICA

$$t - t_0 = \sqrt{\frac{(-a)^3}{\mu}}\left[(e.\,\mathrm{senh}F - F) - (e.\,\mathrm{senh}F_0 - F_0)\right]$$

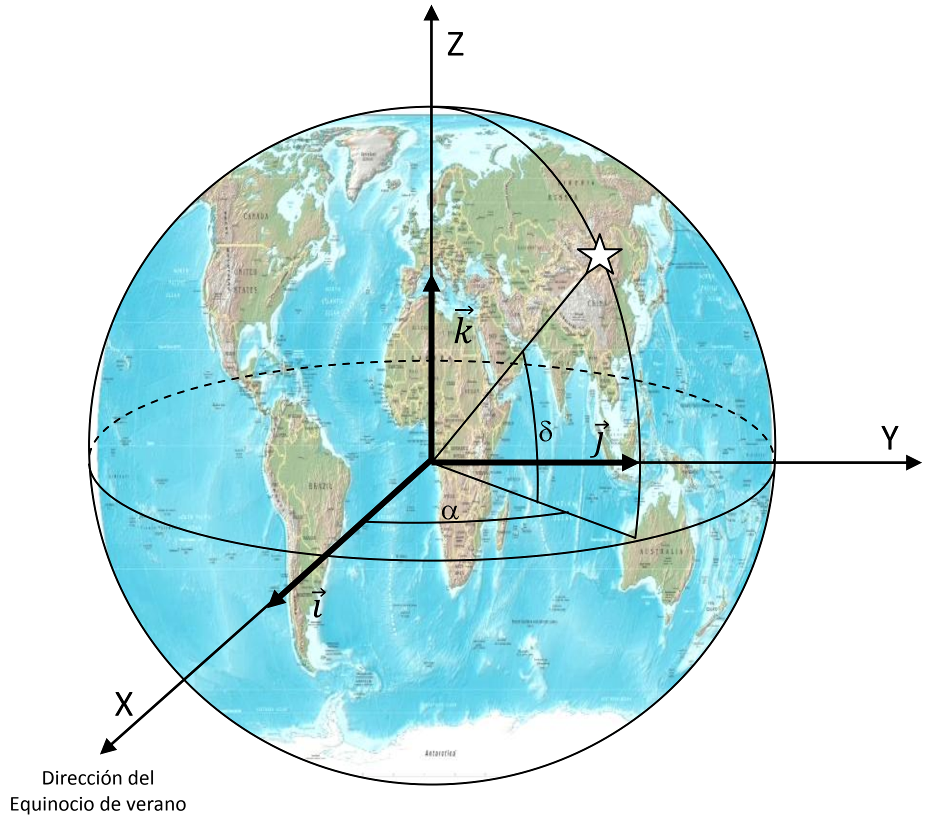
$$\cosh F = \frac{e + \cos\nu}{1 + e\cos\nu}$$

5.3 SISTEMAS DE COORDENADAS y ELEMENTOS ORBITALES CLASICOS

Para poder continuar, es necesario definir dos sistemas de coordenadas, que nos ayudarán a entender los conceptos geométricos que utilizamos para definir la posición de la nave respecto a la tierra.

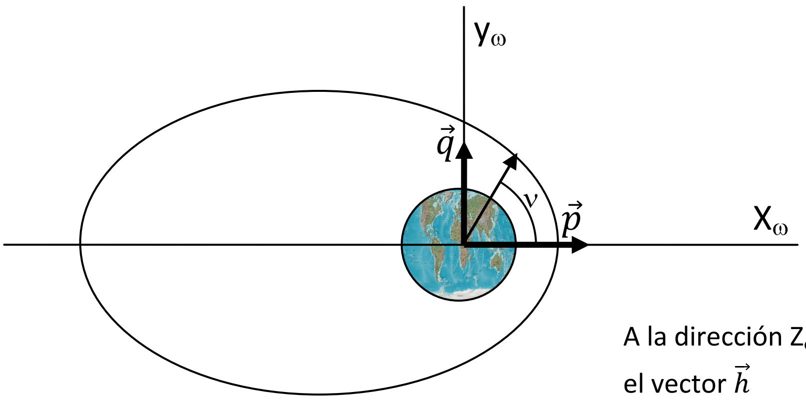
5.3.1 SISTEMA DE CORDENADAS GEOCÉNTRICO-ECUATORIAL

El primero es el sistema geocéntrico-ecuatorial. Es el sistema escogido para referenciar los cuerpos respecto de la tierra. El origen del sistema de coordenadas está situado en el centro de la tierra está formado por el plano sobre el que se describe la órbita de la tierra alrededor del sol (Plano eclíptico) y la dirección perpendicular a dicho plano. La orientación del sistema x-y se establece apuntando el eje x hacia el equinoccio de verano o la dirección del sol.



5.3.2 SISTEMA DE COORDENADAS PERIFOCAL

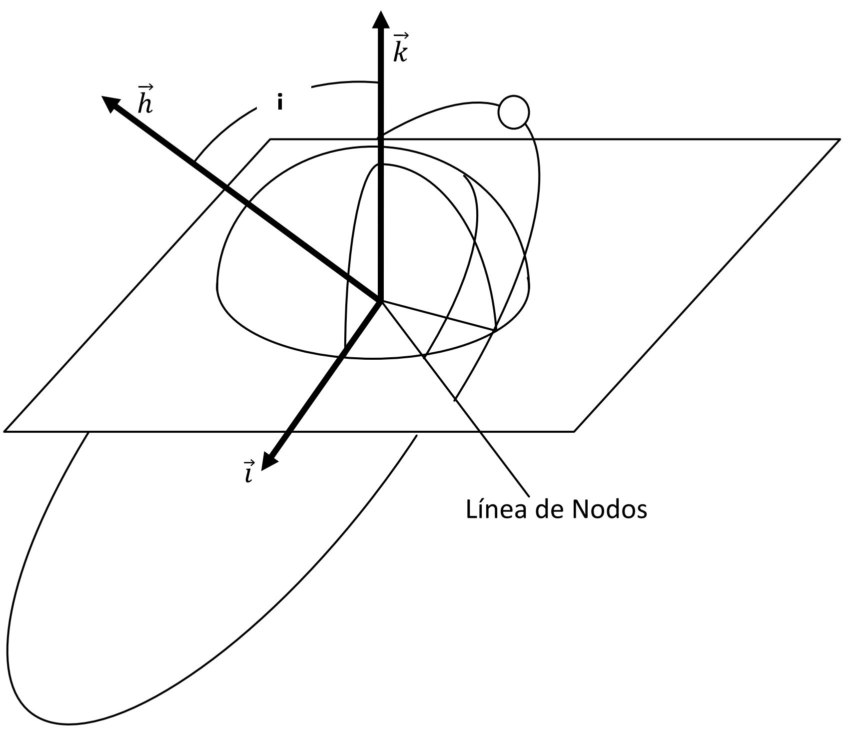
El segundo es el sistema perifocal, denominado así porque establece su origen en el foco del plano de la órbita definida por el satélite. El eje x del plano, apunta hacia el perigeo de la trayectoria. Éste sistema nos permitirá situar el cuerpo en su movimiento orbital y nos ayudará a definir los movimientos en base a las ecuaciones de las curvas cónicas.

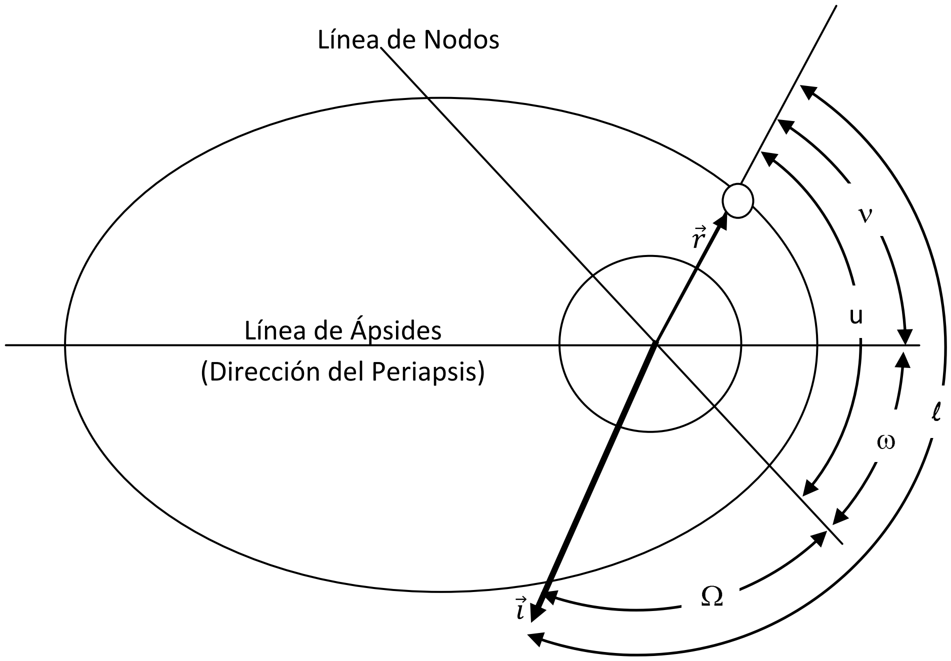


A la dirección Z_ω le corresponde el vector $\vec{h}$

5.4 ELEMENTOS ORBITALES CLÁSICOS

En la siguiente figura, se muestra la intersección entre la órbita descrita por el satélite y el plano de ecuador de la Tierra. A partir de ésta intersección, se definen algunas identidades geométricas importantes que se muestran en el dibujo.





LINEA DE NODOS. Línea de corte entre el plano orbital y el plano de ecuador de la Tierra.

LINEA DE ÁPSIDES. Línea de unión entre el Perigeo y el Apogeo.

Para definir la forma, orientación y tamaño de una órbita, únicamente son necesarios cinco elementos, seis si queremos referenciar en el tiempo la posición del satélite. Clásicamente, se han tomado como válidos los siguientes.

- **SEMI-EJE MAYOR (**a**). Línea de unión entre el perigeo y el apogeo**
- **ECCENTRICIDAD (**e**). Parámetro que define la forma de la curva cónica.**
- **INCLINACIÓN (**i**).** Angulo formado por el plano de la órbita y el del ecuador.
- **LONGITUD DEL NODO ASCENDENTE (Ω).** Angulo formado por la dirección del equinoccio de verano y la línea de nodos.
- **ARGUMENTO DEL PERIAPSIS (ω)** Angulo entre la línea de nodos y la línea de ápsides.
- **TIEMPO DEL PERIAPSIS**. Instante en el que la nave estaba en el periapsis,.

5.5 MANIOBRAS ORBITALES BÁSICAS

Una vez preparados para describir la trayectoria que seguirá nuestra nave después del lanzamiento, debemos centrarnos en la segunda parte del objetivo, que no es otra que mantener la nave dentro de una órbita deseada alrededor de la Luna. Para ello, es necesario conocer los principios que definen las maniobras orbitales básicas.

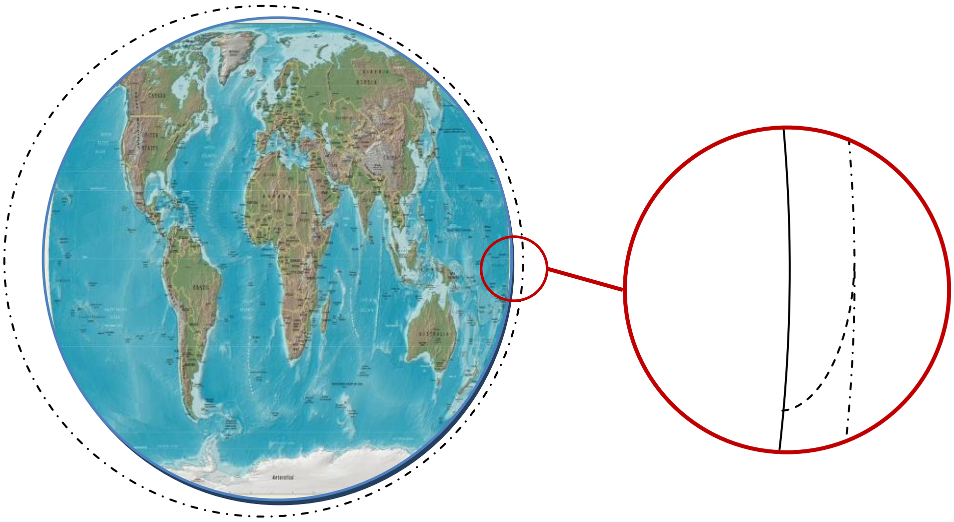
La primera fase de un lanzamiento no es otra que situar el satélite que se quiere lanzar en una órbita cercana a la tierra para posteriormente lanzarlo. Aunque para el algoritmo de cálculo consideraremos a la nave ya en órbita, conviene aclarar algunos conceptos acerca de éste proceso.

5.5.1 Órbitas cercanas a la tierra

Las orbitas cercanas a la tierra, son escogidas cuando nos interesa captar datos desde el espacio como fotografías del terreno.

Estas órbitas, especialmente cuando están tripuladas, están sometidas a ciertas limitaciones. La órbita alrededor de la cual podemos situar el satélite, está limitada a una altura mínima de 100nm para evitar el arrastre atmosférico y a una altura máxima de 300nm para no entrar en los límites de los cinturones de radiación de Van Allen. Ésta altitud, es realmente cercana a la superficie de la tierra, lo suficiente como para entender la precisión que puede ser alcanzada por los satélites de espionaje.

Para situar la nave en ésta órbita, es posible utilizar una maniobra de ascensión directa, calculada para situar la nave en la trayectoria elíptica. Habitualmente se planifica el lanzamiento para situar la nave en el perigeo de la órbita con un ángulo de vuelo $\Phi = 0°$ emplean tres impulsos de inyección, uno para la elevación y dos para corregir los ángulos de posición de la nave.



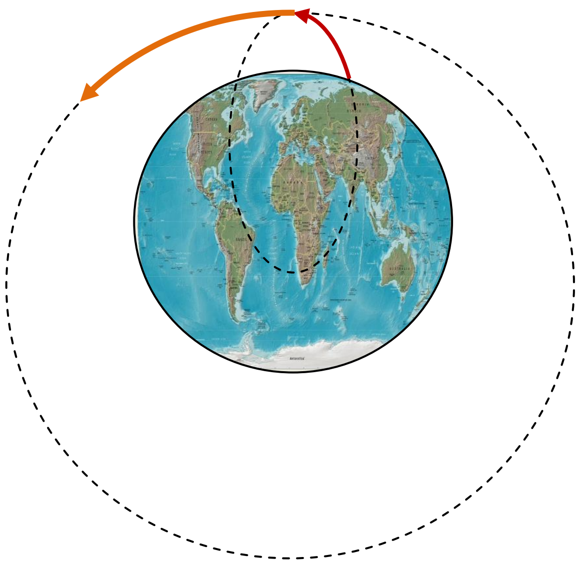
Por último, cabe destacar, que aunque a una distancia elevada de la tierra, su forma achatada no afecta, en una órbita cercana tiene ciertas implicaciones que se deben tener en cuenta. Un es el retroceso de la línea de nodos y la otra una rotación de la línea de ápsides. Estos efectos son producidos por una rotación del plano orbital cuando el satélite está situado en el perigeo o en el apogeo, debido al defecto de masa en sus polos.

5.5.2 Órbitas alejadas de la tierra

Las órbitas a grandes distancias de la tierra, son órbitas escogidas cuando queremos situar nuestra nave lejos del efecto del arrastre atmosférico o deseamos tener una visión global de la tierra. Son las órbitas más adecuadas para los satélites de comunicación.

Dado que al amentar la distancia del satélite a la tierra, también aumenta el periodo de la órbita, podemos llegar a situarnos a una distancia adecuada como para sincronizar el periodo de la órbita, con el periodo de rotación de la tierra. De ésta forma, lograríamos situar un satélite apuntando siempre a la misma superficie de la tierra. Las utilidades en el ámbito de la comunicación para éste tipo de satélites, resultan evidentes.

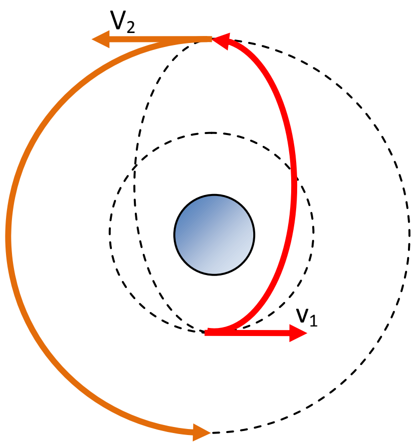
El proceso para el lanzamiento, debe realizarse en dos fases de inyección, una primera que sitúa la nave en una órbita llamada elipse ascendente y una segunda para estabilizar la nave en la órbita externa.



5.5.2 TRANSFERENCIAS DE ÓRBITAS

Para poder situar nuestra nave en una órbita correcta, necesitaremos conocer los métodos para realizar algunas transferencias orbitales que nos permitan situar nuestra nave en la trayectoria deseada.

Uno de los métodos más usados para realizar transferencias entre órbitas circulares, es el método de Hohmann ya que constituye la transferencia más económica en términos energéticos. Se basa incrementar o disminuir la velocidad para entrar en una órbita elíptica que enlace en su perigeo con la órbita de enlace. La siguiente figura puede ayudar a comprender mejor el proceso de la maniobra.



El mismo proceso, podría realizarse de forma inversa.

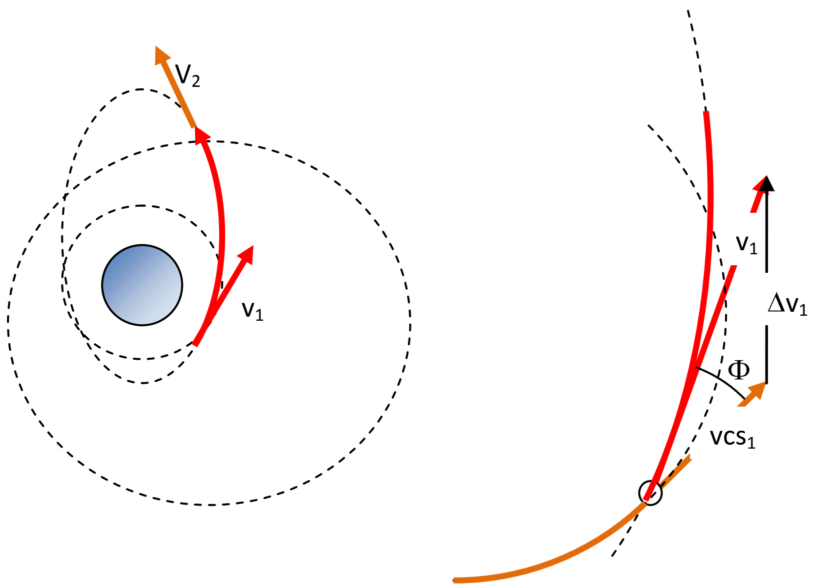
Para calcular el incremento energético, sólo debemos recordar que la energía se mantiene constante a lo largo de la trayectoria y comparar la velocidad en ambas órbitas teniendo en cuanta que el eje mayor de la órbita de transferencia es la suma de ambos radios. De ésta forma llegamos a,

$$a_t = r_1 + r_2 \quad \rightarrow \quad \varepsilon_t = \frac{-\mu}{2(r_1+r_2)} \quad \rightarrow \quad v_1 = \sqrt{2\left(\frac{\mu}{r_1}\right) + \varepsilon_t}$$

$$v_{CS1} = \sqrt{\frac{\mu}{r_1}}$$

$$\Delta v_1 = v_1 - v_{CS1}$$

Existen métodos alternativos para cambiar de órbita cuya única limitación evidente, es que la órbita de transferencia, intercepte en algún punto a ambas curvas, inicial y final.



$$\Delta v_1^2 = v_1^2 + v_{CS1}^2 - 2v_1 v_{CS1} \cos\emptyset_1$$

5.6 TRAYECTORIAS LUNARES

Una vez establecidos los conceptos matemáticos y físicos necesarios para describir el movimiento de los satélites, podemos desarrollar un método simplificado teórico, que nos permita realizar una primera aproximación de la trayectoria que describirá una nave lanzada para orbitar alrededor de la Luna.

Hasta el momento, únicamente nos hemos centrado en un cuerpo afectado por el sistema gravitatorio de la tierra, sin embargo, para un lanzamiento de objetivo lunar, se debe tener en cuenta, el efecto gravitatorio de la Luna. El sistema Tierra-Luna, debido al gran tamaño de la Luna en comparación con el resto de satélites y a las masas móviles de los océanos de la tierra, es un sistema enormemente complejo. Sin embargo, tomaremos en consideración ciertas simplificaciones, que nos permitan realizar una primera aproximación del lanzamiento.

5.6.1 EL SISTEMA TIERRA – LUNA

La Luna es el único satélite natural de la Tierra. Es el cuerpo más cercano y el mejor conocido. La distancia media entre el centro de la Tierra y la Luna es de 384.400 km. Su diámetro (3.474 km) es un cuarto del terrestre, su superficie es una catorceava parte (37.932.330 km^2), y su volumen alrededor de una cincuentava parte (21.860.000.000 km^3).

Los movimientos que describe la Luna con respecto a la tierra, son dos. De traslación y de rotación. También describe un tercero alrededor del sol.

El primero, **de traslación**, hace completar a la Luna una vuelta alrededor de la Tierra aproximadamente en unos 28 días. Si la Tierra no rotara, se vería sería la Luna cruzando la bóveda celeste de oeste a este durante dos semanas, y luego hasta desaparecer para hacerse visible en el lado opuesto del Globo, sin embargo, debido al movimiento de rotación de la Tierra el efecto que se produce se trata de un retraso de la Luna de alrededor 12°51’ cada día respecto de su posición del día anterior.

El segundo, **de rotación** se trata de un giro sobre un eje con una inclinación de 88,3° con respecto al plano de la elíptica. Su duración, está sincronizada con el movimiento de traslación siempre apunta hacia la tierra el mismo hemisferio de la Luna por lo que sólo es posible observar una cara desde la tierra.

La trayectoria que describe la Luna alrededor de la Tierra es, como cabía esperar elíptica y por tanto, su velocidad no es uniforme. Dado que la rotación lunar sí lo es se produce una Libración en longitud que permite ver un poco más de la superficie lunar al Este y al Oeste.

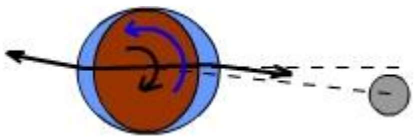
Dado que el plano de la órbita lunar está inclinado respecto a la Eclíptica unos 5º también se produce una Libración en latitud que permite ver alternativamente un poco más allá del polo Norte o del Sur. Por ambos movimientos el total de superficie lunar vista desde la Tierra alcanza un 59% del total.

Cada vez que la Luna cruza la eclíptica, si la Tierra y el Sol están sensiblemente alineados (Luna llena o Luna nueva) se producirá un eclipse lunar o un eclipse solar.

La órbita de la Luna es especialmente compleja. En primer lugar, la Luna está suficientemente lejos de la Tierra (384.400 km en promedio) para que la fuerza de gravedad ejercida por el Sol sea significativa. La línea de nodos de la Luna, descrita por el plano de la eclíptica y el perifocal de la órbita Lunar, gira alrededor del equinoccio de verano hasta completar una vuelta en 18,6 años. Como el semieje mayor de la órbita elíptica no está fijo la inclinación de la órbita varía entre 5º y 5º 18'. Para calcular la posición de la Luna con exactitud harían falta tener en cuenta cientos de términos.

La Luna, también se aleja unos cuatro centímetros al año de la Tierra y poco a poco va frenando la rotación terrestre de forma que en un futuro lejano los eclipses totales de Sol no podrán de producirse debido a que la superficie proyectada de la Luna será insuficiente para tapar el disco solar. Aunque teóricamente dicha separación debería prolongarse hasta aumentar en 47 días el ciclo lunar de modo similar a lo que ocurre en el sistema Plutón-Caronte, la evolución futura de nuestro Sol puede trastocar ésta evolución ya que al convertirse en una estrella gigante roja dentro de varios miles de millones de años, la proximidad de su superficie al sistema Tierra-Luna haga cerrarse a la órbita lunar el límite de Roche, provocando que la gravedad terrestre destruirá la Luna convirtiéndola en unos anillos similares a los de Saturno.

Si definimos a la tierra y a la Luna como puntos en el espacio con masa, en realidad, el movimiento de traslación de la Luna no se realiza en torno a la Tierra, sino que ambas giran en torno al centro de masas del sistema. Debido al enorme tamaño de la tierra, la gravedad ejercida por la Luna es distinta en cada punto siendo mucho mayor en la zona más próxima que en la alejada. El gradiente gravitatorio producido ejerce una deformación sobre la superficie líquida deformando la tierra y dándole el aspecto de un huevo. Este es el fenómeno que produce las mareas haciendo que las aguas suban y bajen dos veces al día (en los puntos más cercano y más alejado de la Luna).



Un efecto asociado es que las mareas frenan a la Tierra en su rotación (pierde energía debido a la fricción de los océanos con el fondo del mar), y dado que el sistema Tierra-Luna tiene que conservar el momento angular, la Luna lo compensa alejándose, actualmente, 38 mm cada año, como han demostrado las mediciones láser de la distancia, posibles gracias a los retro-reflectores que los astronautas dejaron en la Luna.

5.6.2 Trayectorias simples lunares.

Aunque el cálculo preciso de una trayectoria lunar, únicamente puede ser realizado a partir de integraciones numéricas debido a la cantidad de perturbaciones que influyen como la presión solar, perturbaciones solares, el campo gravitatorio lunar y cientos de ellas más, la mejor manera de optimizar un proceso de integración numérica es partir de unas condiciones iniciales lo más adecuadas posibles para arrancar el proceso de iteración.

Para poder obtener una solución analítica que nos permita arrancar un método iterativo de cálculo, es necesario establecer un modelo simplificado sobre las hipótesis que estableceremos a continuación.

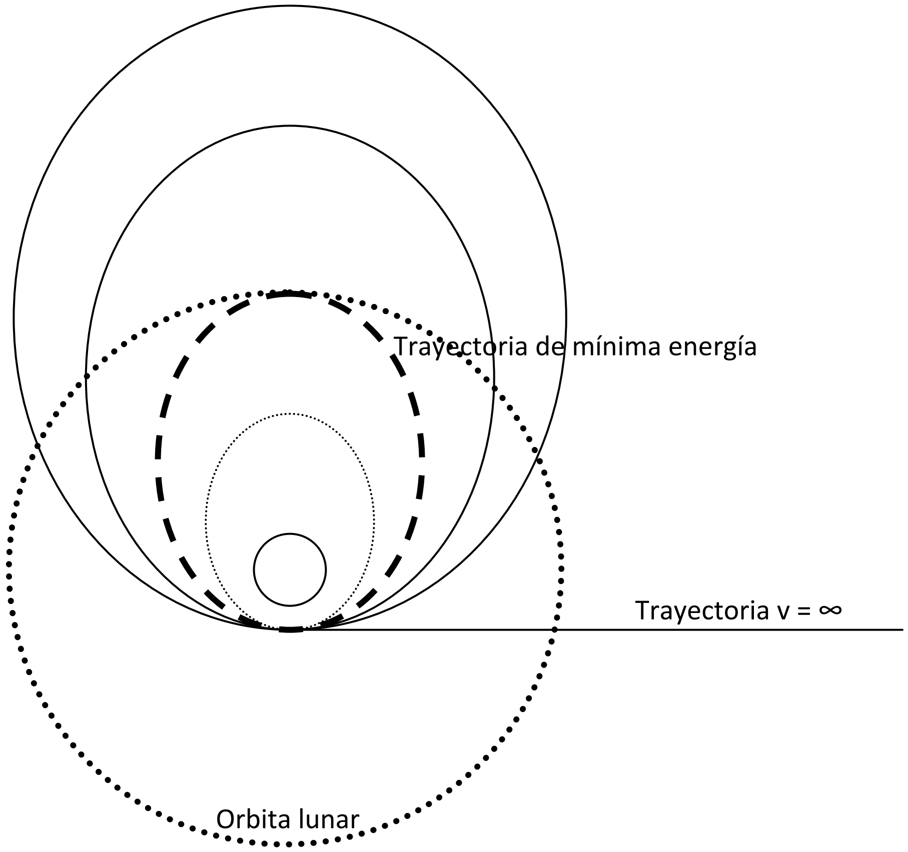
5.6.2.1 Modelo simplificado ignorando la gravedad de la Luna.

- Supondremos la órbita lunar como circular con un radio de 384.400 km. Dado que la excentricidad media actual es de 0,0549, no introducirá grandes errores.
- Supondremos la gravedad de la luna como nula.
- Supondremos la trayectoria de la nave, coplanaria con la órbita Lunar.

Debemos tener en cuenta, que al no considerar el campo gravitatorio de la Luna, no es posible calcular la trayectoria descrita por un satélite en su órbita y por tanto, éste modelo sólo será útil, si deseamos impactar sobre la Luna.

Cálculo de la mínima energía en el lanzamiento.

Suponiendo que el lanzamiento se produce desde el perigeo (Φ_0 = 0º) en función de la velocidad de escape, podemos definir diferentes trayectorias para alcanzar la órbita lunar, correspondientes a diferentes modelos cónicos.



Si observamos la figura, podemos apreciar diferentes trayectorias en función de su velocidad de escape. La línea rayada en negrita, se trata de la velocidad mínima de impulsión. A ésta velocidad nuestra trayectoria será tangente a la de la órbita lunar, alcanzando así nuestro objetivo con el mínimo gasto energético. La velocidad mínima de impulsión, debe ser aquella que genere una trayectoria elíptica con su eje mayor igual al radio de la órbita lunar. La mínima velocidad de inyección es por tanto 10.82 km/s.

***Distancia perdida por errores en la velocidad de inyección*.**

Si cometemos un error en la velocidad de vuelo, nuestra trayectoria variará, por ejemplo, si la velocidad es demasiado alta, la curva será más abierta e interceptaremos la trayectoria lunar en un punto anterior al planificado. Sin embargo, dado que nuestra velocidad es mayor, el tiempo de vuelo es menor y por tanto, el instante en el que interceptaremos con la trayectoria lunar, la luna estará también en una posición anterior a la planificada y compensará nuestro error en el lanzamiento. Así que si nuestro error de velocidad, se ve compensado, se puede decir que el error en el lanzamiento depende casi por completo de un error en el ángulo de vuelo (fly-path-angle) y de nuestra posición inicial en el instante del lanzamiento.

5.6.2.2 APROXIMACION PATCHED-CONIC

Para poder tener en cuenta la gravedad de la Luna, podemos realizar una aproximación razonable suponiendo que un cuerpo deja de estar influenciado por la gravedad de la tierra y empieza a estarlo por la gravedad de la Luna en un punto concreto de forma inmediata. Evidentemente, esto no sucede así pero si calculamos, dentro de los radios de acción de la gravedad de ambos campos, un punto en el que las fuerzas de la gravedad terrestre y lunar sean iguales, podremos realizar una buena aproximación, suficiente para calcular una solución inicial que arranque nuestro modelo iterativo. (No obstante, para el modelo de regreso a la tierra no se trata de una buena aproximación ya que el no tener en cuenta el momento de transición en el que ambas campos actúan sobre la nave, podría introducir grandes errores en el destino de aterrizaje de la nave sobre la tierra).

Esferas de influencia de Laplace

Si imaginamos el campo gravitatorio de ambos cuerpos como esferas cerradas donde la influencia gravitatoria sigue siendo relevante, obtendremos una ecuación simplificada que defina la gravedad ejercida por cada astro en función del radio. Igualando ambas fuerzas gravitatorias, podemos encontrar una distancia intermedia, donde ambos campos queden anulados por su efecto contradictorio. Dicho punto fue definido por Laplace según la siguiente ecuación.

$$R_s = D\left(\frac{m_{moon}}{m_{Å}}\right)^{2/5}$$

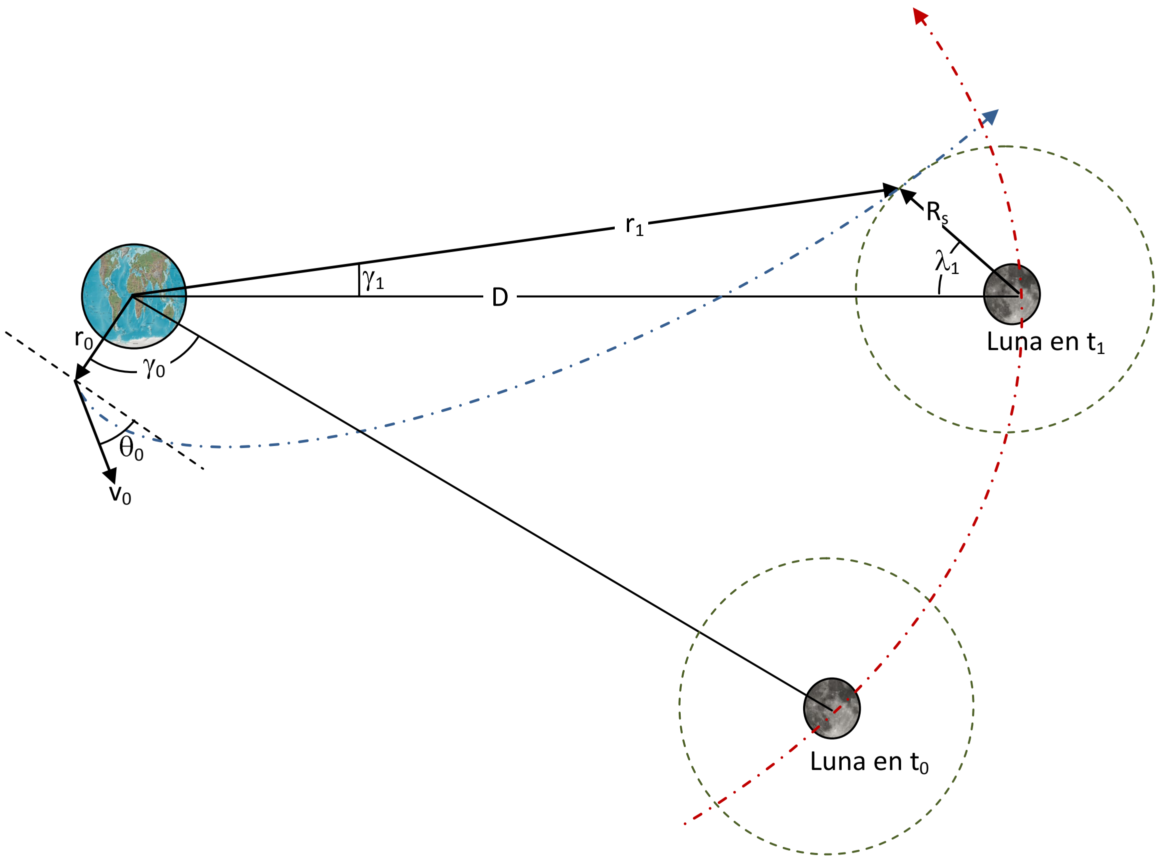
Donde D es la distancia de la tierra a la Luna. Por tanto el R_s = 66.300 km. Aproximadamente 1/6 de la distancia entre la tierra y a Luna.

5.6.2.3 PROCEDIMIENTO PARA EL CÁLCULO DE CONDICIONES DE LLEGADA

En base a las ecuaciones planteadas anteriormente y a las simplificaciones del modelo, podemos describir un método para calcular las condiciones de llegada de una nave al entorno de la órbita Lunar y las ecuaciones de la trayectoria descrita por el cuerpo una vez dentro de la órbita lunar. Dado que el objetivo que se persigue es un método que pueda ser computarizado, vamos a realizar a descripción como pasos secuenciales.

A. <u>Especificación de las condiciones iniciales</u>.

Debemos imponer las condiciones iniciales del lanzamiento. Para entender mejor la geometría de la trayectoria que describirá nuestra nave, podemos observar el dibujo siguiente, Donde se describe en trazos discontinuos la trayectoria que realizará la nave tras el lanzamiento.



Las condiciones iniciales que estableceremos serán

r_o - Posición inicial de la nave (Respecto al centro de la tierra)

v_0 – velocidad inicial

Φ_0 – Angulo inicial de vuelo

λ_1 – Posición angular de la nave en el punto de llegada

Nótese que para poder dar una solución inequívoca, debemos introducir una condición inicial que nos de información del vector de llegada. De forma práctica, el ángulo que formará la posición de la nave en el momento de llegada respecto a la recta de unión entre los centros de la Luna y la tierra λ_1.

B. Calculo de condiciones de llegada a R_s.

A partir de éstas condiciones iniciales y utilizando las ecuaciones de conservación de la energía y el momento cinético, podemos calcular de forma analítica, las condiciones de llegada de la aeronave deducir para nuestro modelo simplificado.

Del dibujo anterior, deducimos por geometría que,

$$\mathrm{sen}(\gamma 1) = \left(\frac{Rs}{r1}\right).\mathrm{sen}(\lambda 1)$$

Utilizando la ley de cosenos, deducimos que,

$$\boxed{r_1 = \sqrt{D^2 + R_s^2 - 2DR_s\cos(\lambda_1)}}$$

Ec. 9

A partir de la conservación de la energía,

$$\boxed{\varepsilon = \frac{v_0^2}{2} - \frac{\mu}{r_0}}$$

Ec. 10

A partir de la conservación del momento angular,

$$\boxed{h = R_0.V_0.\cos(\phi_0)}$$

Ec. 11

A partir de las leyes de Kepler, calculamos el tiempo de vuelo. Para ello, en primer lugar debemos definir los parámetros que caracterizan la curva cónica definida por la aeronave,

$$2p = \frac{2h^2}{\mu} \qquad \text{(Latus rectum)}$$

$$a = \frac{-\mu}{2\varepsilon} \qquad \text{(Semieje mayor)}$$

$$e = \sqrt{1 - \frac{p}{a}} \qquad \text{(Excentricidad)}$$

Una vez calculada la excentricidad podemos aplicar las ecuaciones para calcular el tiempo de vuelo,

$$\cos(\upsilon_0) = \frac{p - r_0}{r_0 e} \qquad \text{True anomally}$$

$$\cos(\upsilon_1) = \frac{p - r_1}{r_1 e} \qquad \text{True anomaly}$$

$$\cos(E_0) = \frac{e + \cos(\upsilon_0)}{1 + e.\cos(\upsilon_0)} \qquad \text{Excentric anomally}$$

$$\cos(E_1) = \frac{e + \cos(\upsilon_1)}{1 + e.\cos(\upsilon_1)} \qquad \text{Excentric anomally}$$

$$M = \left(E_1 - e.\mathrm{sen}E_2\right) \qquad \text{Mean anomally}$$

$$n = \sqrt{\frac{a^3}{\mu}} \qquad \text{Mean motion}$$

$$t_1 - t_0 = \sqrt{\frac{a^3}{\mu}}.\left[\left(E_1 - e.\mathrm{sen}E_1\right) - \left(E_0 - e.\mathrm{sen}E_0\right)\right] \qquad \text{Ley de Kepler}$$

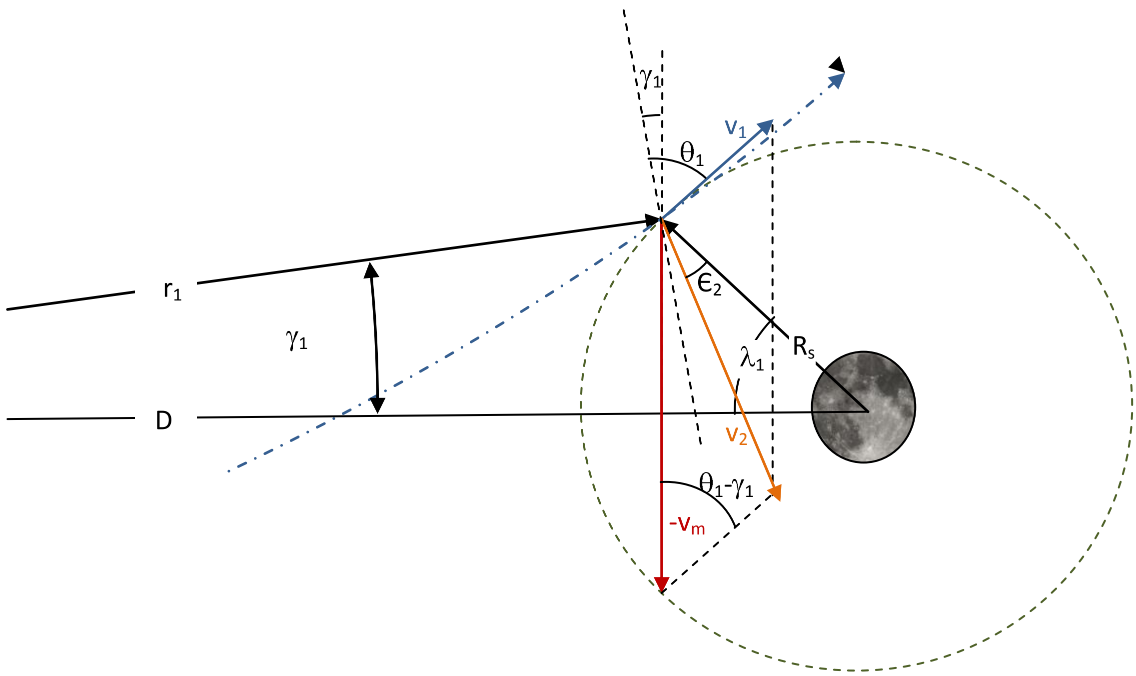
Por último, una vez calculado el tiempo que necesitará la nave para recorrer la trayectoria, podemos calcular la posición de la Luna en el momento de llegada de la aeronave,

$$\gamma_0 = \upsilon_1 - \upsilon_0 - \gamma_1 - \varpi_m(t_1 - t_0)$$

C. Trayectoria cónica respecto de la Luna

Ahora, ya conocemos las condiciones en las que llegará la aeronave al área de influencia de la Luna. Sin embargo, nos interesa conocer el comportamiento de la nave dentro de la influencia del campo gravitatorio Lunar, para determinar comportamiento que tendrá la nave en su recorrido final y comprobar que es consistente con la naturaleza de la misión que deseamos realizar (Impacto Lunar, Aterrizaje…).

A partir del siguiente dibujo, podemos determinar la posición y la velocidad de nuestra nave con respecto de un sistema de coordenadas perifocal con su origen en la luna para, de ésta forma, determinar los parámetros energéticos respecto del campo gravitatorio lunar.



Teniendo en cuenta los valores calculados de la nave respecto a la tierra de v_1 y la velocidad de la Luna respecto a ésta v_m podemos calcular la velocidad de la aeronave tomando como origen la propia Luna.

Por tanto tenemos que,

$$r_2 = R_s$$

$$v_2 = \sqrt{v_1^2 + v_m^2 - 2v_1 v_m \cos(\Phi_1 - \Upsilon_1)}$$

$$\epsilon_2 = \operatorname{arcsen}\left(\frac{v_m}{v_2}\cos(\lambda_1) - \frac{v_1}{v_2}\cos(\lambda_1 + \gamma_1 - \phi_1)\right)$$

Una vez calculadas la velocidad y el radio respecto de la luna, ya podemos calcular los parámetros energéticos y los parámetros característicos de la trayectoria.

$$\varepsilon_m = \frac{v_2^2}{2} - \frac{\mu_m}{R_S}$$

$$h_m = R_s . v_2 . \operatorname{sen}(\epsilon_2)$$

$$2p_m = \frac{2h_m^2}{\mu_m}$$

$$a_m = \frac{-\mu_m}{2\varepsilon_m}$$

$$e_m = \sqrt{1 - \frac{p_m}{a_m}}$$

Una vez definida la cónica del movimiento alrededor de la Luna, ya podemos predecir el comportamiento de la nave con respecto a ésta y corregir su trayectoria, a partir de maniobras orbitales para cumplir con los objetivos de la misión.

6 DESARROLLO DEL ALGORITMO DE CÁLCULO

6.1 OBJETIVO DEL ALGORITMO

Una vez establecidos los principios físico-matemáticos necesarios para abordad el objetivo del proyecto, vamos a centrarnos en el desarrollo de un algoritmo que nos permita calcular las condiciones iniciales óptimas para un lanzamiento que sitúe un satélite no tripulado alrededor de la luna.

El objetivo será establecer una metodología de cálculo y un esqueleto de programa capaz de evolucionar en el tiempo para abordar paso a paso cada una de las complejidades del sistema hasta contemplar todos los detalles que sean precisos. En definitiva, se trata de crear un programa que establezca la base simplificada de un algoritmo capaz de evolucionar hasta convertirse en una herramienta suficientemente precisa como para ser utilizada en el cálculo o en la corrección de los parámetros iniciales de navegación de un lanzamiento real.

6.2 METODOLOGÍA Y ESTRUCTURA DEL PROGRAMA

Lo primero que debemos hacer antes de comenzar a escribir código, es definir exactamente la estrategia que deseamos seguir. Para ello, vamos a estructurar el programa en cinco pasos fundamentales. Éstos pasos, aunque en una primera versión serán tremendamente sencillos, podrían llegar complicarse todo lo que se quisiera, pero sin dejar de cumplir el objetivo fundamental para el que están descritos.

Aunque el objetivo fundamental, sería preparar la misión en el espacio tridimensional, por el momento vamos a centrarnos en el modelo bidimensional, suponiendo que la órbita de nuestro lanzamiento es coplanaria con la órbita lunar.

Los pasos a seguir pueden ser enumerados de la siguiente manera.

1. Establecimiento y lectura de condiciones iniciales de la misión.
2. Calculo de condiciones iniciales para una primera aproximación del lanzamiento.
3. Calculo de una primera aproximación del tempo de vuelo y el ángulo de salida.
4. Algoritmo iterativo de simulación para obtención del tiempo de vuelo y el ángulo de salida correcto.
5. Algoritmo iterativo de simulación para realizar un mapeado de v0 y optimizar el incremento de velocidad.
6. Presentación de resultados y condiciones iniciales para misión optima.

1. Establecimiento y lectura de condiciones iniciales de la misión.

En éste primer punto, deben ser definidas las variables que van a condicionar la misión. La lectura de estas variables, aunque deben ser preestablecidas, deberían ser leídas de uno o varios ficheros de condiciones iniciales con el objetivo de modificar el contenido de éstos si fuese necesario integrar algún otro programa que necesitase realizar cálculos en base a otra serie de parámetros para definir los que son necesarios par a el programa.

Las condiciones iniciales sobre las que partirá nuestro programa serán.

- Radio inicial del lanzamiento (r_0)
- Angulo de velocidad (θ_0)
- Fecha y hora del lanzamiento.
- Ángulo Ω de la órbita lunar en dicho instante.
- Radio de la órbita lunar circular objetivo (R_L)
- Precisión del radio para el resultado

Nótese, que la especificación del ángulo Ω es en realidad redundante, ya que se podría calcular en base a las efemérides de la Luna para la fecha indicada. No obstante, por una cuestión de simplificación hemos preferido indicarlo para evitar un cálculo que tendremos en cuenta para una posible mejora de programa.

2. Calculo de condiciones iniciales para una primera aproximación del lanzamiento.

El cálculo de las condiciones iniciales consiste en una primera aproximación teórica en base a un modelo simplificado, que nos dé una primera velocidad inicial con la que arrancar el método iterativo que utilizaremos para obtener la velocidad más optima para el lanzamiento.

Ésta primera aproximación, persigue únicamente, obtener una velocidad que no sitúe en un lanzamiento válido para nuestro propósito por lo que aunque puede ser mejorada para acelerar el método, en un principio estará basa en un sistema tremendamente simplificado suficiente para alcanzar el objetivo perseguido.

3. Calculo de una primera aproximación del tempo de vuelo y el ángulo de salida.

El orbitador lunar, nos proporciona la trayectoria descrita por la nave y por la luna, en un intervalo de tiempo determinado. Para poder validar el lanzamiento, es necesario determinar la duración del vuelo para situar a la nave en la posición precisa en el instante adecuado. Para conseguir éste propósito necesitamos realizar un primer proceso iterativo que corrija el tiempo de vuelo hasta detectar la duración exacta en la que la nave está situada en la posición objetivo.

Para arrancar dicho método, utilizaremos una primera estimación del tiempo de vuelo en base al modelo simplificado.

Al margen del funcionamiento del orbitador, calcular el tiempo de vuelo es determinante ya que nos dará información acerca de la posición inicial que debe tener nuestra nave en el instante del lanzamiento para poder encontrarse con la órbita lunar a partir del ángulo inicial γ_0 o desde el punto de vista de los parámetros orbitales, del ángulo de la línea de nodos con el equinoccio de verano (Ω).

4. *Algoritmo iterativo de simulación para obtención del tiempo de vuelo y el ángulo de salida correcto.*

Una vez calculada la primera aproximación del tiempo de vuelo y de la velocidad inicial v0, podemos llamar al orbitador lunar para comprobar la posición real, o al menos, más precisa, que tendrá nuestra nave a partir de los datos iniciales. Cabe esperar que la primera aproximación no sea del todo correcta y la posición de nuestra nave no sea la deseada por lo que arrancaremos un proceso iterativo de prueba y error, que nos ayude a calcular la duración de vuelo correcto y la posición inicial de la nave necesaria para el encuentro con la órbita Lunar.

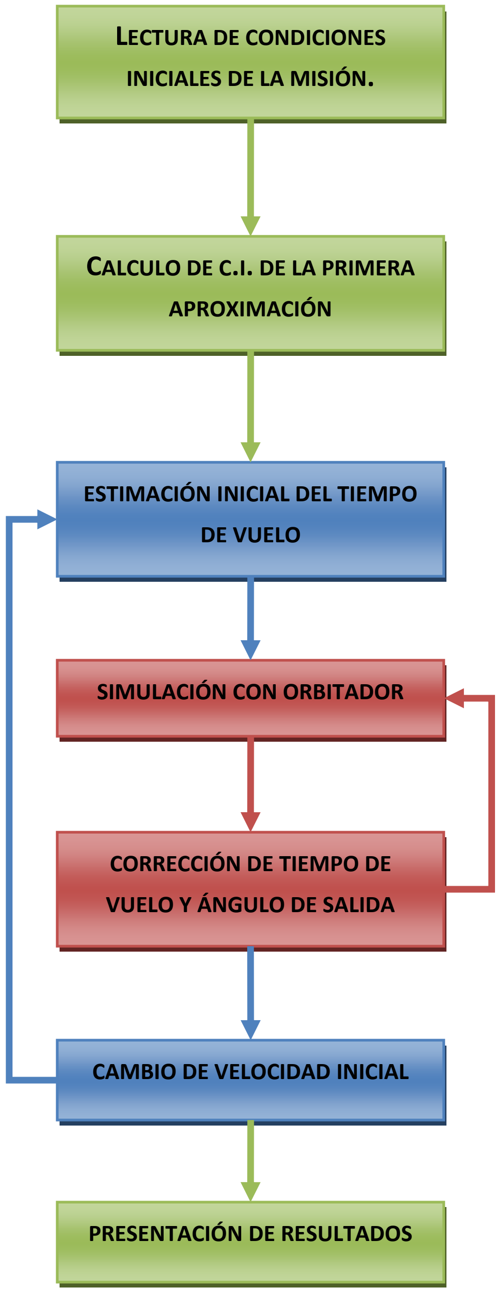
5. *Algoritmo iterativo de simulación para realizar un mapeado de v0 y optimizar el incremento de velocidad.*

Aunque hayamos encontrado unas condiciones iniciales válidas para nuestro lanzamiento, es posible que no sean las más óptimas por lo que a partir de un mapeado de velocidades iniciales, trataremos de repetir los pasos anteriores para una gama de velocidades válidas que nos permitan analizar las diferentes posibilidades para el lanzamiento con el fin de escoger aquella que implique un menor gasto energético.

6. *Presentación de resultados y condiciones iniciales para misión optima.*

Por último, debemos presentar los resultados del análisis y definir de forma clara los parámetros que se consideran más adecuados para el lanzamiento.

En la siguiente figura, se muestra un esquema de la metodología general seguida por el programa.



6.3 HIPÓTESIS INICIALES PARA EL DESARROLLO DEL MODELO

Siguiendo el esquema anterior, el primer paso a realzar se trata de la adquisición de las condiciones iniciales que determinarán la situación inicial de nuestra nave, así como la naturaleza de la misión.

6.3.1 OBJETIVO DE LA MISIÓN

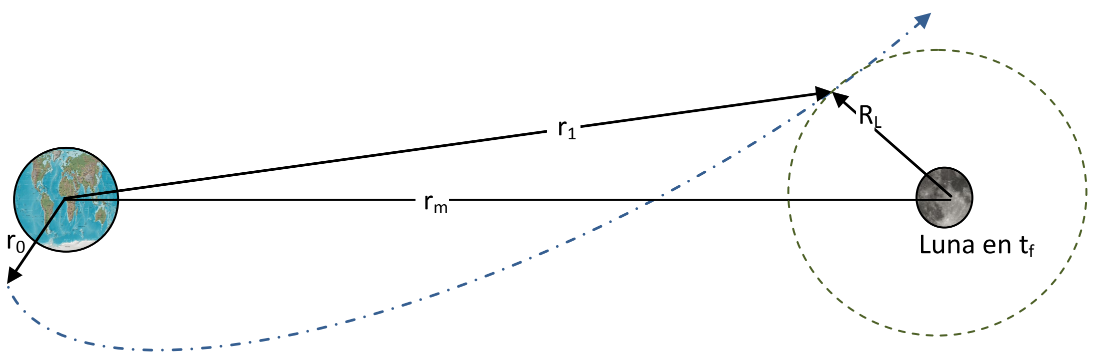
En pocas palabras, el objetivo de la misión, consistirá en lanzar una nave para situarla en una órbita circular alrededor de la Luna. Hemos elegido utilizar una órbita circular para poder simplificar la estrategia de cálculo, pero con leve modificaciones sería posible adaptar la misión a cualquier tipo de órbita o eligiendo una órbita lo suficientemente pequeña, planear incluso un impacto.

El hecho de que nuestra nave deba orbitar alrededor de la luna, establece cual debe ser el estado final de la nave y condicionará por tanto las condiciones iniciales que debe tener el lanzamiento. Es estado final de que debe tener nuestra nave al concluir su trayectoria, queda definido esencialmente por tanto por dos parámetros, la posición y la velocidad.

Posición.

Dado que la órbita Lunar es circular, la posición relativa de nuestra nave con respecto a la luna debe ser igual al radio de la órbita lunar escogida R_L . Por tanto, en la instante final del lanzamiento debe cumplirse que,

$$|\vec{r_1} + \vec{r_m}| = R_L$$



Velocidad

De la misma manera, la velocidad que debe tener nuestra nave, debe ser consecuente con la velocidad de una órbita circular. El módulo de esta velocidad, puede calcularse como,

$$|\overrightarrow{V_{CS}}| = \sqrt{\frac{\mu_m}{R_L}}$$

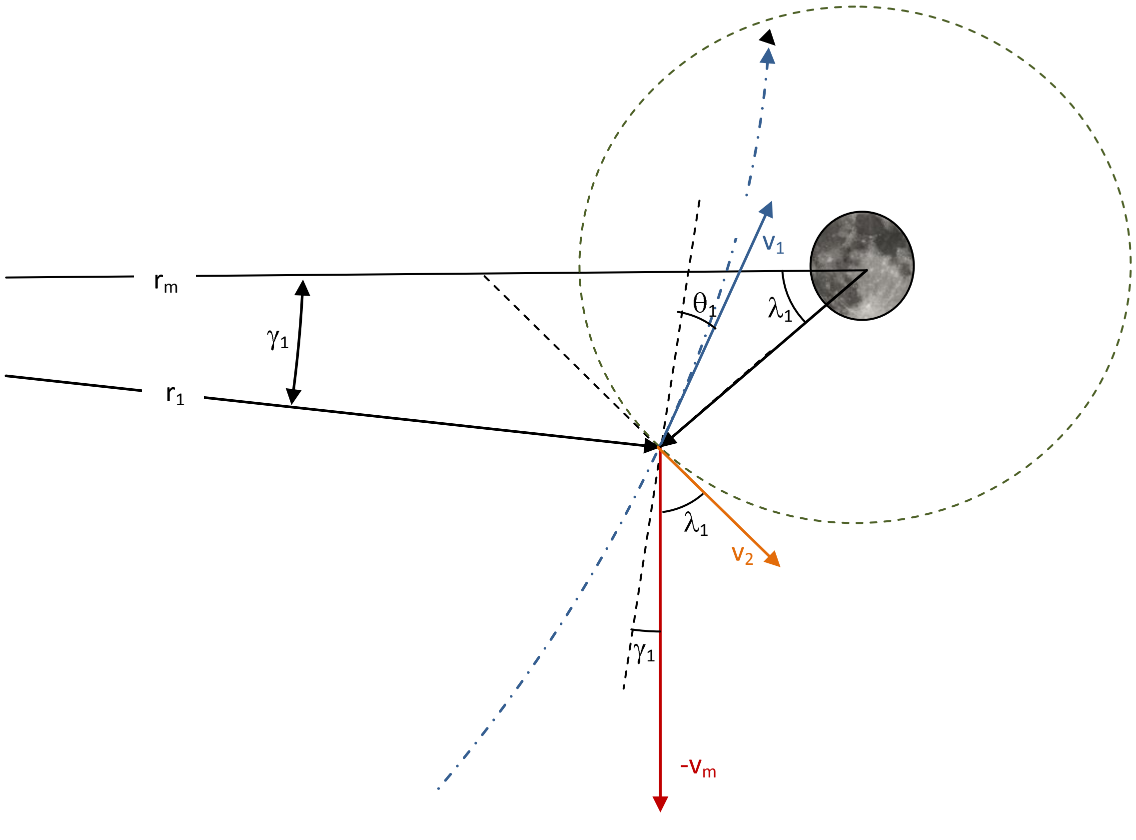
Siendo su dirección siempre perpendicular a la dirección del radio y su sentido horario, para aprovechar la dirección de la velocidad del lanzamiento.

Por otro lado, en el instante final del lanzamiento, se cumple que,

$$|\overrightarrow{v_1} - \overrightarrow{v_m}| = \overrightarrow{v_2}$$

Por tanto, el lanzamiento ideal óptimo sería aquel que cumpliese que,

$$\overrightarrow{v_2} = \overrightarrow{v_{CS}}$$



En la figura anterior, se representa ésta situación donde la suma de velocidades, es en el radio de la órbita circular exactamente igual a la velocidad necesaria para describir la órbita.

Del dibujo anterior deducimos que,

$$\mathrm{v}_1 = \sqrt{\mathrm{v}_2^2 + \mathrm{v}_\mathrm{m}^2 - 2\mathrm{v}_2\mathrm{v}_2\cos\lambda_1}$$

$$v_2 = v_{cs}$$

$$\cos\theta_1 = \frac{\mathrm{v}_\mathrm{m}\cos\gamma_1 - \mathrm{v}_2\cos(\gamma_1 + \lambda_1)}{\mathrm{v}_1}$$

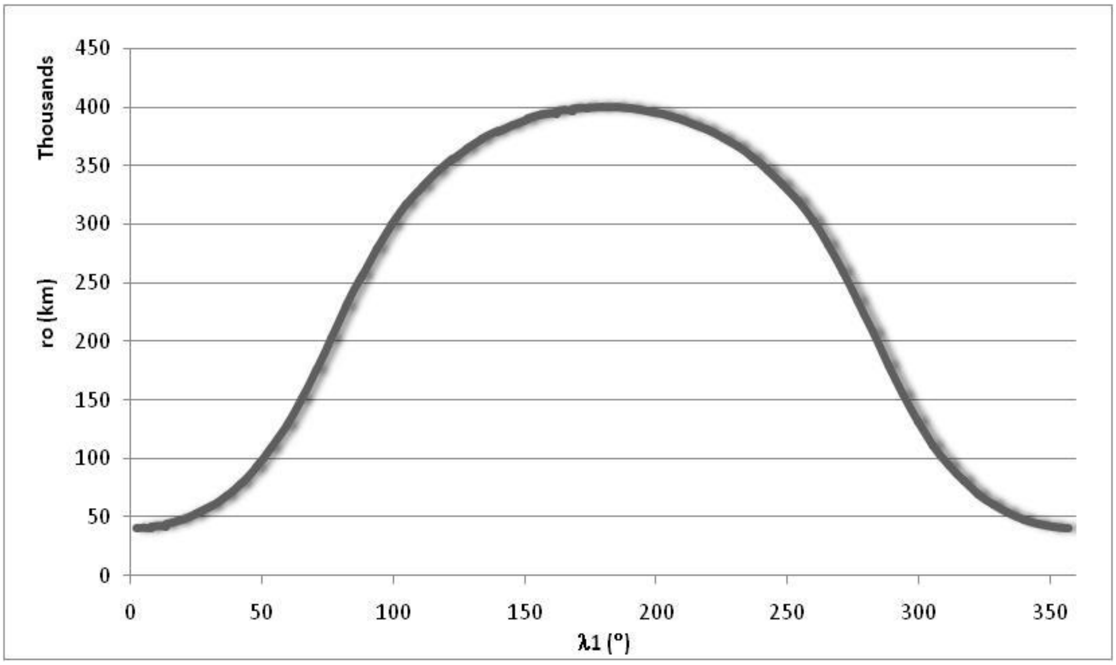
Y a partir de éste punto, encontramos los parámetros energéticos que definen la trayectoria siendo

$$\varepsilon = \frac{\mathrm{v}_1^2}{2} - \frac{\mu}{\mathrm{r}_1}$$

$$\mathrm{h} = \mathrm{r}_1.\mathrm{v}_1.\cos(\theta_1)$$

Aunque en principio, parece una opción viable y calculable, estos parámetros en realidad, condicionan la órbita que describe nuestra nave y por tanto los ángulos de salida y los radios iniciales válidos.

Si representamos en todo el rango posible de ángulos de llegada λ_1 el radio de la órbita en el perigeo, bajo el supuesto de ignorar la gravedad de la Luna, descubrimos que el radio mínimo inicial necesario para describir una órbita válida es de al menos 6 veces el diámetro de la tierra.



Éste radio es demasiado grande para una órbita de salida por lo que tendríamos que situar la nave a una distancia válida y después impulsar la nave para modificar la trayectoria.

$$\boxed{|\vec{r_1}+\vec{r_m}|=R_L}$$

Ec. 12

Sin embargo, ya que debemos asumir el hecho de que necesitaremos una impulsión más para alcanzar nuestro objetivo, lo más prudente sería realizar la impulsión a una distancia de la luna igual al radio de la órbita deseada. Por tanto, **podemos simplificar nuestro objetivo cumpliendo simplemente la ecuación 12.**

Una vez situados en dicha posición, analizaremos la velocidad para calcular el impulso necesario para adaptarnos a la velocidad de la órbita circular necesaria.

6.3.2 SITUACIÓN INICIAL DE LA NAVE. ÓRBITA DE PARTIDA

El resto de condiciones iniciales, van a depender de la situación de nuestra nave en el momento del lanzamiento. Aunque el propósito de éste programa, es precisamente definir algunos de ellos, es necesario establecer alunas invariables.

Nuestra nave, estará situada en una órbita alrededor de la luna, con una inclinación que en principio estableceremos exactamente igual a la inclinación de la órbita lunar. En el momento del lanzamiento, asumiremos que la nave se encuentra justo en la línea de nodos por lo que su ángulo ω será igual a 0°. Sin embargo, el ángulo Ω vendrá definido por el tiempo de vuelo y las condiciones de la Luna en el instante del lanzamiento.

Dado que el ángulo Ω de la trayectoria no coincide con el de la órbita de la Luna, el hecho de que la inclinación de ambas órbitas sea el mismo, no implica que ambas trayectorias sean coplanarias. Sin embargo, realizaremos todo el algoritmo de cálculo bajo éste supuesto sobre la proyección de ambas órbitas sobre el plano ecuatorial para más adelante corregir ésta imprecisión.

Las condiciones iniciales sobre las que partirá nuestro programa serán.

- Radio inicial del lanzamiento (r_0)
- Angulo de velocidad (θ_0)
- Fecha y hora del lanzamiento.
- Ángulo Ω de la órbita lunar en dicho instante.

No es necesario indicar, que sería posible modificar éstas condiciones para adaptarlas a supuestos que fuesen más interesantes, sin embargo, se han establecido estas como ejemplo de aplicación siendo consideradas como lógicas en la planificación de un lanzamiento.

Como futuras ampliaciones de programa, cabría contemplar, la variación del ángulo de salida con el objetivo de minimizar aún más el gasto energético necesario.

6.4 CALCULO DE CONDICIONES DE PRIMERA APROXIMACIÓN

6.4.1 HIPOTESIS PARA LA PRIMERA APROXIMACIÓN DE LA VELOCIDAD INICAL

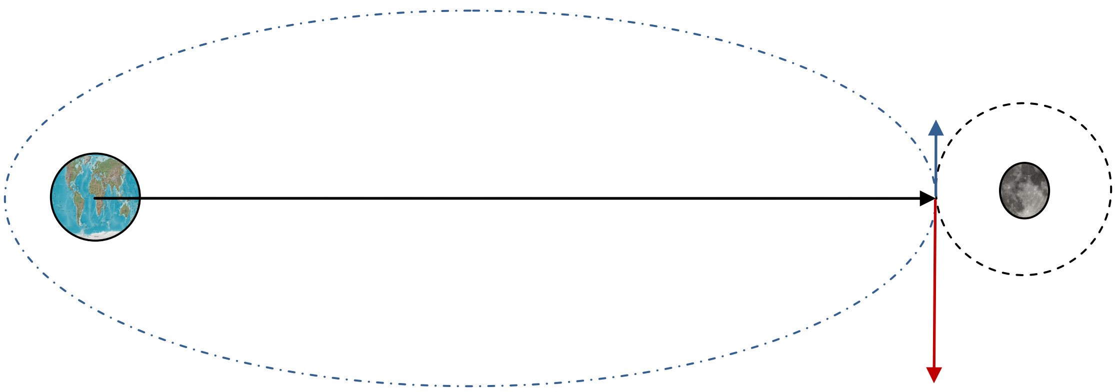
Para que el algoritmo iterativo sea convergente y lo haga de una manera rápida, debemos alimentarlo con una primera aproximación lo suficientemente buena como para evitar iteraciones inútiles que puedan desencadenar en un proceso divergente. Si la velocidad inicial no es suficiente para el lanzamiento, dado que el objetivo de nuestro proceso iterativo es asegurar que $|\vec{r_1}+\vec{r_m}| = R_L$ se cumpla, si realizamos el lanzamiento con una velocidad demasiado baja, el proceso iterativo no podría converger nunca, porque no existiría una solución válida para esa velocidad. Por ello, debemos asegurarnos la velocidad inicial del lanzamiento es suficiente para llevar a la nave hasta la esfera descrita por el radio de la órbita circular determinada.

Para obtener una primera solución simplificada vamos a analizar el sistema más sencillo que asegura que nuestro sistema tiene una velocidad suficiente. Si nos centramos únicamente en la órbita descrita por la nave sin tener en cuenta la acción gravitatoria de la luna, la máxima distancia que alcanzará nuestra nave, será el radio en el apogeo de la trayectoria. Tomando la ecuación de la cónica para el caso del perigeo donde ν=0°, podemos escribir ésta distancia como,

$$r_a = \frac{p}{1-e} = a(1+e)$$

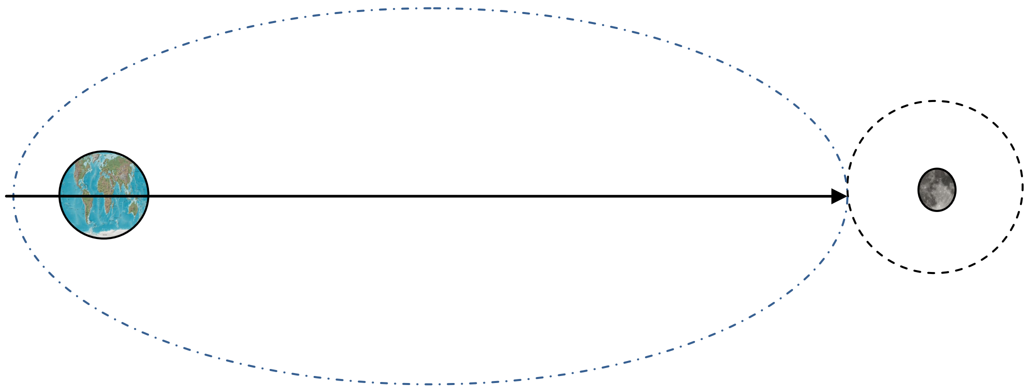
Y su velocidad,

$$v_p = \sqrt{2\left(\varepsilon + \frac{\mu}{r_p}\right)}$$



Como se puede observar en la figura, la velocidad en el perigeo es perpendicular al radio y por tanto, su resultante con respecto a la luna sería únicamente la suma de sus módulos.

Una buena primera aproximación sería una velocidad igual a la de la luna menos la velocidad de la órbita circular, ya que situaría a la nave justo en órbita con un solo impulso pero como vimos en el capítulo anterior, esto solo es posible para radios iniciales demasiado grandes, así que la velocidad inicial que utilizaremos será la que nos permita llegar a la luna con una velocidad igual a cero partiendo de nuestra posición inicial.



Por tanto, según las ecuaciones anteriores tenemos que,

$r_1 = d - R_L$ $\qquad v_1 = 0$ $\qquad \varepsilon = \frac{-\mu}{d-R_L}$ $\qquad h = 0$

Despejando la velocidad inicial necesaria de la ecuación de la energía, obtenemos que,

$v_0 = \sqrt{2\left(\varepsilon + \frac{\mu}{r_0}\right)}$ $\qquad \theta_0 = 0$

Dado que ésto es sólo una aproximación, cabe pensar que en un modelo más complejo que tuviese en cuanta la gravedad de la luna, la velocidad no sea suficiente para alcanzar dicha distancia, sin embargo, dado que el efecto gravitatorio de la luna es atrayente, contribuirá en acelerar nuestra nave por lo que la velocidad será seguramente mayor de la necesaria.

6.4.2 HIPOTESIS PARA LA PRIMERA APROXIMACIÓN DEL TIEMPO DE VUELO

Aunque hemos utilizado una aproximación para obtener un módulo de velocidad suficiente para alcanzar la distancia que requiere el encuentro con la Luna, dado que el objetivo de la misión no es alcanzarla, sino situarnos en su órbita circular, será necesario calcular el tiempo de vuelo así como el ángulo inicial del lanzamiento.

Para ello, vamos a utilizar la misma aproximación, es decir, no tendremos en cuenta el efecto gravitatorio de la luna y las condiciones iniciales que tendremos en cuenta serán por tanto,

r_0, v_0, θ_0, y r_1

Siguiendo la metodología descrita en el apartado 5.6, debemos aplicar las siguientes ecuaciones,

1. $\cos\lambda_1 = \dfrac{D^2 + R_s^2 - r_1^2}{2DR_s}$

2. $\varepsilon = \frac{v_0^2}{2} - \frac{\mu}{r_0}$

3. $h = R_0.V_0.\cos(\phi_0)$

4. $v_1 = \sqrt{2\left(\varepsilon + \dfrac{\mu}{r_1}\right)}$

5. $\cos(\phi_1) = \dfrac{h}{r_1.V_1}$

6. $2p = \dfrac{2h^2}{\mu}$

7. $a = \dfrac{-\mu}{2\varepsilon}$

8. $e = \sqrt{1 - \dfrac{p}{a}}$

9. $\cos(\upsilon_0) = \dfrac{p - r_0}{r_0 e}$

10. $\cos(\upsilon_1) = \dfrac{p - r_1}{r_1 e}$

11. $\cos(E_0) = \dfrac{e + \cos(\upsilon_0)}{1 + e.\cos(\upsilon_0)}$

12. $\cos(E_1) = \dfrac{e + \cos(\upsilon_1)}{1 + e.\cos(\upsilon_1)}$

13. $t_1 - t_0 = \sqrt{\dfrac{a^3}{\mu}}.[(E_1 - e.\mathrm{sen}E_1) - (E_0 - e.\mathrm{sen}E_0)]$

14. $\gamma_0 = \upsilon_1 - \upsilon_0 - \gamma_1 - \varpi_m(t_1 - t_0)$

Una vez aplicadas, tendremos loa datos suficientes para llamar al orbitador y simular el primer lanzamiento.

Velocidad teórica insuficiente

Dado que los datos que estamos proponiendo, son únicamente una primera aproximación en la que no tenemos en cuenta el efecto gravitatorio de la Luna, es posible que la velocidad inicial propuesta nos proporcione una trayectoria con una distancia al apogeo menor que la distancia especificada en r_1. Ésta situación, no tiene porqué ser así en el orbitador ya que la atracción de la Luna, puede ser capaz de acelerar nuestra nave para permitirle alcanzar la distancia necesaria. En éste caso, dado que teóricamente es imposible alcanzar una distancia mayor que la del apogeo, al establecer como condición inicial la distancia r_1, encontraremos ecuaciones irresolubles o con soluciones en el plano imaginario para el cálculo de v_1 y θ_1. Al no tener soluciones válidas para éstas variables no podremos hacer una estimación del tiempo de vuelo y por tanto, no podremos arrancar el proceso iterativo para el cálculo del tiempo de vuelo correcto.

Para evitar éste problema. Necesitamos añadir una condición en el cálculo que nos permita disminuir el radio final del lanzamiento hasta encontrar una solución válida para que nos proporcione una estimación del tiempo de vuelo con la que arrancar el proceso de iteración independientemente del resultad teórico del modelo simplificado.

Para ello, debemos tener en cuenta las ecuaciones de cálculo de v_1 y θ_1.

$$v_1 = \sqrt{2\left(\varepsilon + \frac{\mu}{r_1}\right)}$$

$$\cos(\phi_1) = \frac{h}{r_1 . V_1}$$

Para que éstas ecuaciones no tengan solución, basta con que el valor del operando interior de la raíz sea negativo, o que el coseno sea mayor que la unidad. Si esto ocurre, debemos ajustar el radio final del lanzamiento, al máximo viable para la trayectoria descrita, es decir, la distancia al apogeo.

Por tanto debemos incluir en nuestra operación de cálculo, la condición siguiente.

$$\text{Si } \left(\varepsilon + \frac{\mu}{r_1} < 0\right) \text{ ó } \left(\frac{h}{r_1 . V_1} < 1\right) \quad \rightarrow \quad r_1 = \frac{p}{1 - e}$$

De esta manera, aseguramos una solución inicial para arrancar el proceso iterativo.

6.4.3 FUNCIONES ASOCIADAS

Para computarizar este proceso vamos a definir en el programa dos funciones. Una para el cálculo de la velocidad inicial, y otra para la estimación del tiempo de vuelo. La razón de ésta división es que probablemente podamos complicar alguno de los dos algoritmos sin necesidad de tocar el otro. Si quisiésemos por ejemplo tener en cuenta la gravedad de la luna a partir de las ecuaciones descritas en la aproximación Patched –Conic, el proceso sería muy sencillo para el cálculo del tiempo de vuelo, pero no tan evidente para la primera aproximación de la velocidad por lo que sería lógico incluir dicha modificación únicamente en el segundo algoritmo.

Las funciones definidas para realizar la aproximación son las siguientes:

float faissm (init_par); //First aproximation Init ship speed Mathematics

La función faissm, devuelve el valor de la velocidad inicial a través de una variable tipo float. Es necesario alimentar la función con los valores de las condiciones iniciales utilizando la estructura init_par compuesta por los siguientes elementos.

float r0,	Radio de la posición inicial de la nave
float th0,	Angulo inicial para la velocidad de impulsión
float omega,	Angulo entre la línea de nodos de la órbita lunar y el eje x del plano ecuatorial
float rfmo,	Radio de la órbita circular objetivo
float acc;	Precisión requerida para la posición final del lanzamiento
int year, month, day, hour, minute, second	Fecha del lanzamiento

ship_state faftm (init_par,float); //First aproximation Flight time and ship state mathematics

La función faftm devuelve el estado inicial de la nave en el momento de la impulsión a partir de la estructura ship_state. Hay que alimentarla con las condiciones iniciales a través de una estructura tipo *init_par* y con la velocidad inicial del lanzamiento con una variable tipo *float.*

6.5 PROCESO ITERATIVO PARA CORRECCIÓN DEL TIEMPO DE VUELO

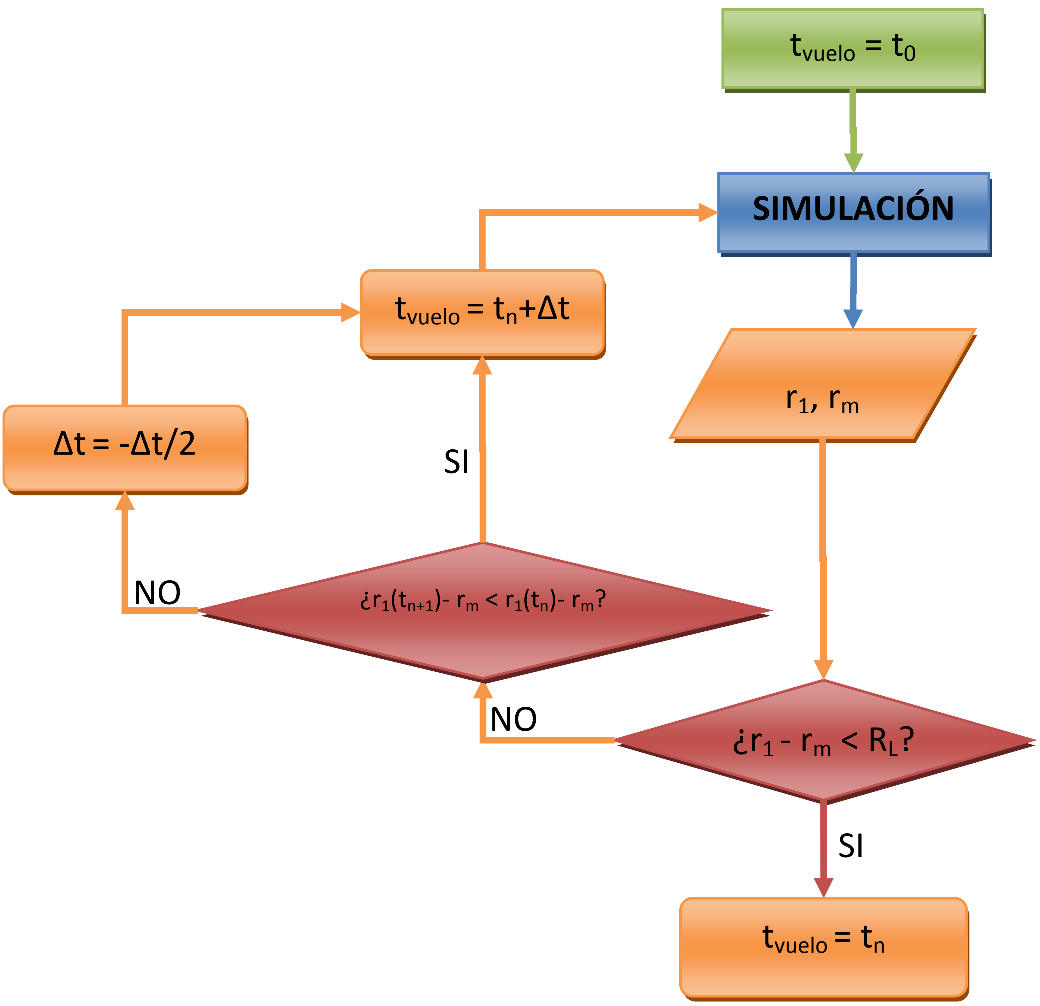
Para una obtención de un tiempo de vuelo más preciso, es necesario utilizar un orbitador que tenga en cuenta las complejidades del sistema tierra luna. En nuestro caso utilizaremos el orbitador *esmoprop v1.0.*

El objetivo del proceso es realizar llamadas al orbitador de forma iterativa hasta que la distancia entre la nave y la Luna, sea prácticamente igual al radio de la órbita necesaria, o expresado de forma matemática hasta que,

$$\left| |\overrightarrow{v_1}| - |\overrightarrow{v_m}| \right| < \epsilon$$

Donde ϵ es un error tan pequeño como nosotros queramos.

A partir del siguiente diagrama, puede verse claramente la dinámica de la iteración



6.5.1 CASOS ESPECIALES DEL PROCESO ITERATIVO

Aunque el concepto de la iteración es muy sencillo, dado que la velocidad inicial es constante y el ángulo Ω también el hecho de partir de una aproximación introduce la posibilidad de que se den una serie de casos concretos que deben ser contemplados y corregidos ya que de lo contrario podrían llevarnos a una iteración infinita.

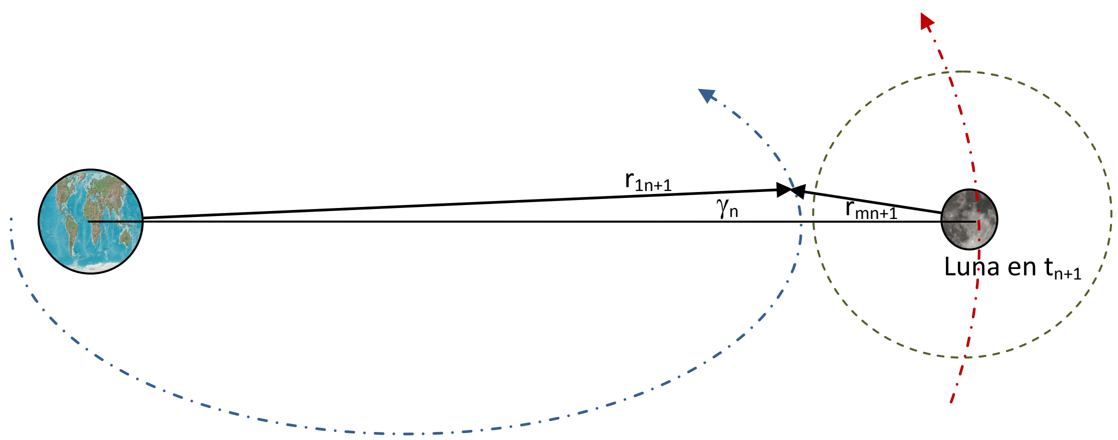
Los casos que deben tratarse son los siguientes,

- Velocidad inicial insuficiente para alcanzar la órbita circular de la luna
- Angulo inicial erróneo, demasiado desfase entre la tierra y la luna
- Posición de la nave demasiado alejada en el momento del encuentro.

1. *Velocidad inicial insuficiente*

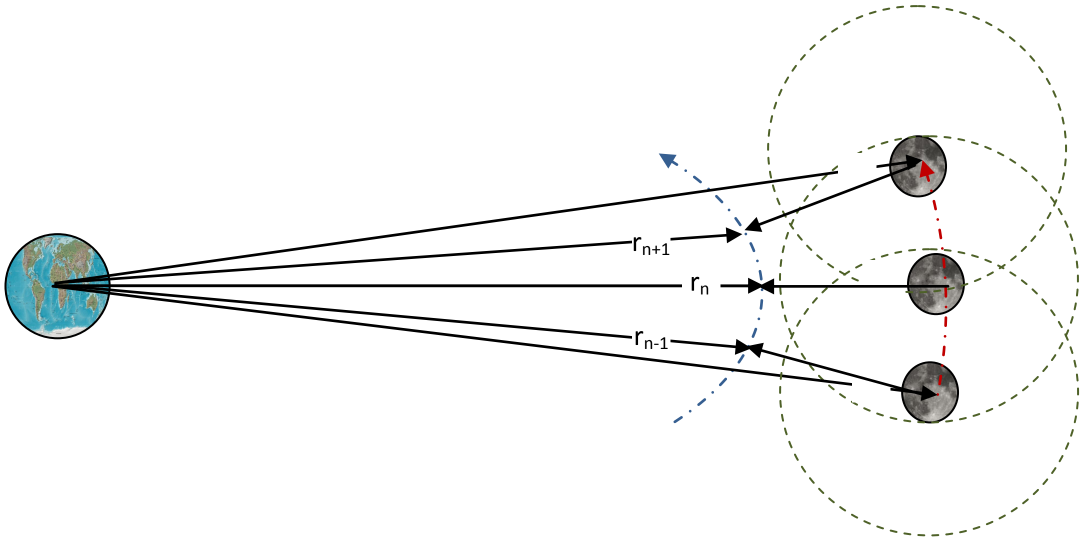
Cualquiera de los casos anteriores, nos llevará a un proceso de iteración infinito y aunque es posible realizar una corrección sobre las siguientes iteraciones, antes debemos establecer un método que nos permita detectar que nos encontramos en un proceso de iteración divergente.

Para entender cómo funciona el proceso iterativo, vamos a analizar el primero de los casos, el más evidente, una velocidad inicial de lanzamiento insuficiente para alcanzar la distancia que requiere un contacto con la órbita circular buscada.

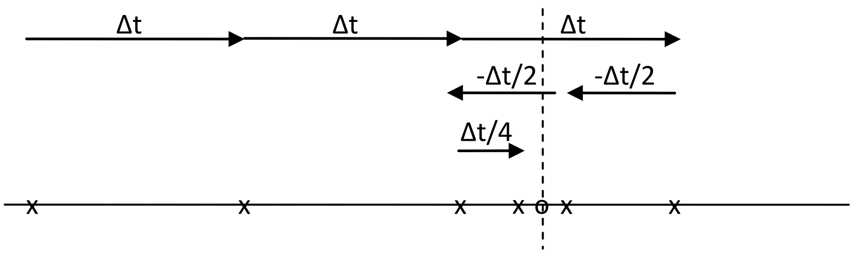


Al contrario que en el resto de los casos anteriores, no tomaremos ésta situación como un lanzamiento erróneo que debe ser corregido para seguir iterando sino como un indicador de fin de iteración ya que el objetivo de esta etapa es encontrar el tiempo de vuelo adecuado para una velocidad inicial determinada, y no corregir esa velocidad inicial. Por tanto, si detectamos ésta situación, finalizaremos el proceso iterativo y enviaremos al programa principal un indicado de que el la velocidad enviada no es suficiente para alcanzar la distancia mínima al objetivo.

Para buscar un método de detección de una iteración infinita debemos entender cómo funciona de forma gráfica el proceso iterativo.



Durante el proceso iterativo, la modificación del incremento de tiempo, se reduce cuan do se produce un cambio de signo. Si el valor de R_L-r_1 cambia de signo, quiere decir que hemos aumentado demasiado el tiempo y por tanto debemos reducirlo a un ritmo más lento.



Observando la figura anterior, es fácil entender que si nuestra velocidad es insuficiente para alcanzar la distancia mínima que requiere un contacto con la órbita circular, tanto en t_{n+1} como en t_{n-1} la distancia a la luna será menor que en t_n y por tanto habrá un cambio de signo constante reduciéndose de forma constante el incremento de tiempo, sin llegar a alcanzar nunca un radio que cumpla la condición de salida de la iteración. Entraremos por tanto en un proceso infinito.

Un indicador de que esta situación es el valor del incremento de t, ya que podemos establecer una condición que nos haga salir de la iteración cuando el incremento se reduzca por debajo de un valor determinado.

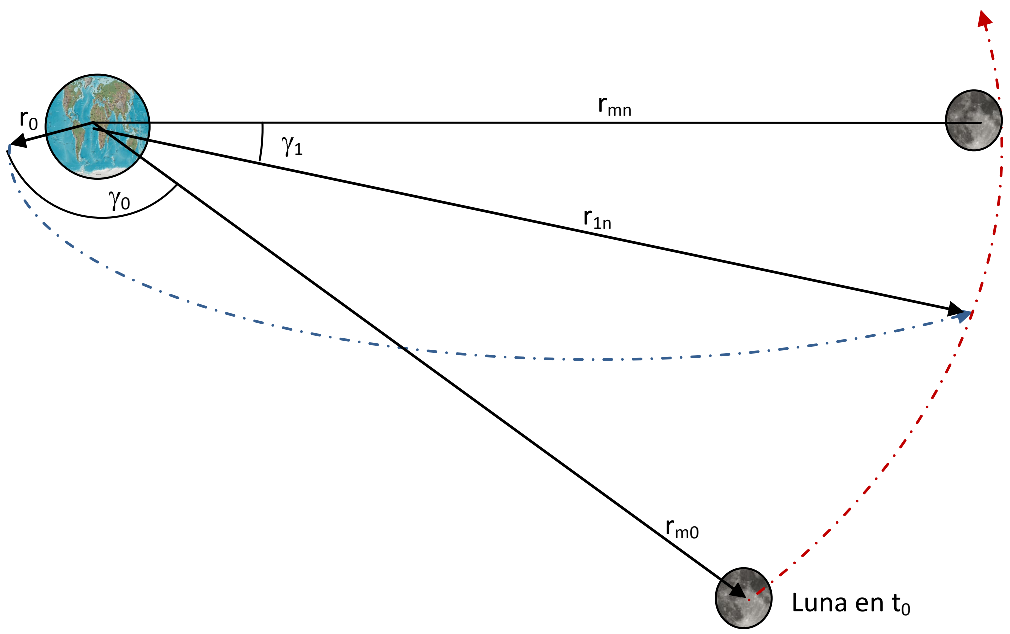
Sin embargo, ésta condición no es suficiente para detectar que nuestra velocidad es insuficiente ya que podríamos encontrarnos en cualquiera de los casos que analizaremos con detalle más adelante. Debemos añadir una condición más que nos asegure que es el defecto de velocidad el que está produciendo ésta iteración infinita. Para ello, simplemente comprobaremos que nuca hemos llegado a superar la distancia requerida.

Dado que se ha cumplido la primera de las condiciones, significa que se han producido cambios de signo es decir, de tendencia en la evolución del r_2 y por tanto si no estamos en ninguno de los casos que explicaremos a continuación, la velocidad del lanzamiento no es suficiente para alcanzar la órbita.

2. *Angulo inicial erróneo*

Dado que las condiciones iniciales de la primera iteración están basadas en una simplificación del modelo, es muy probable que introduzcan errores significativos, siendo especialmente débil el cálculo del tiempo de vuelo.

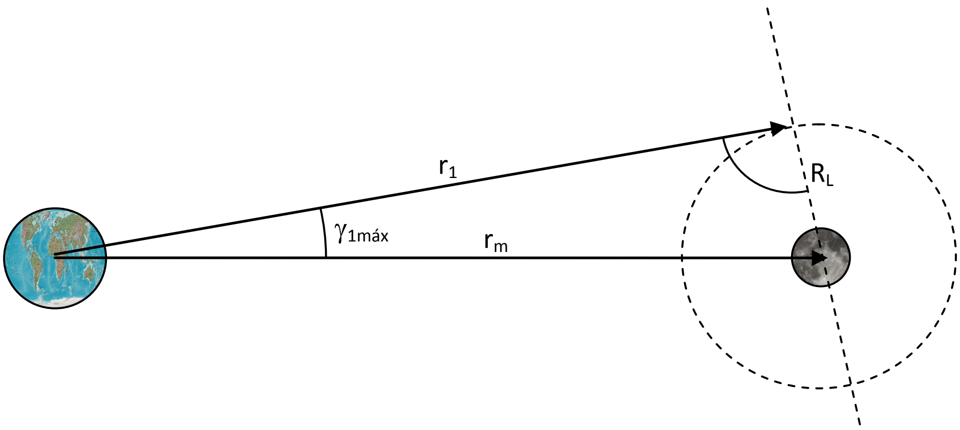
Como la atracción del campo gravitatorio lunar contribuye a acelerar la nave, cuanto menor sea la velocidad inicial del lanzamiento, mayor será el error en el tiempo de vuelo. Aunque éste error es corregido por la iteración anterior, el cálculo del ángulo inicial γ_0 está basado en ésta primera estimación y puede darse el caso de que el error sea suficiente como para que las trayectorias se encuentren en una posición equivocada e incluso que no lleguen a tocarse.



Como podemos observar en la figura, la forma de detectar esta situación y corregir el error es bastante sencilla. Para ello sólo debemos medir el ángulo γ_1 y corregir el ángulo del lanzamiento inicial γ_0 con ese mismo valor de forma que en la siguiente iteración, la nave y la luna, quedarán perfectamente alineadas con la tierra.

Luna en t_1

El valor que tomaremos como límite para decidir si el ángulo γ_1 es demasiado grande, es el que formado por una recta que pasa por el origen de coordenadas y es tangente a la órbita que se desea describir, ya que se trata del ángulo máximo que permite el contacto de la nave con la órbita lunar para el radio devuelto por el orbitador.



$$\gamma_{1max} = \text{arctg}\left(\frac{\mathrm{r_m}}{\mathrm{r_1}}\right)$$

La corrección del error la realizamos añadiendo la siguiente condición a las iteraciones.

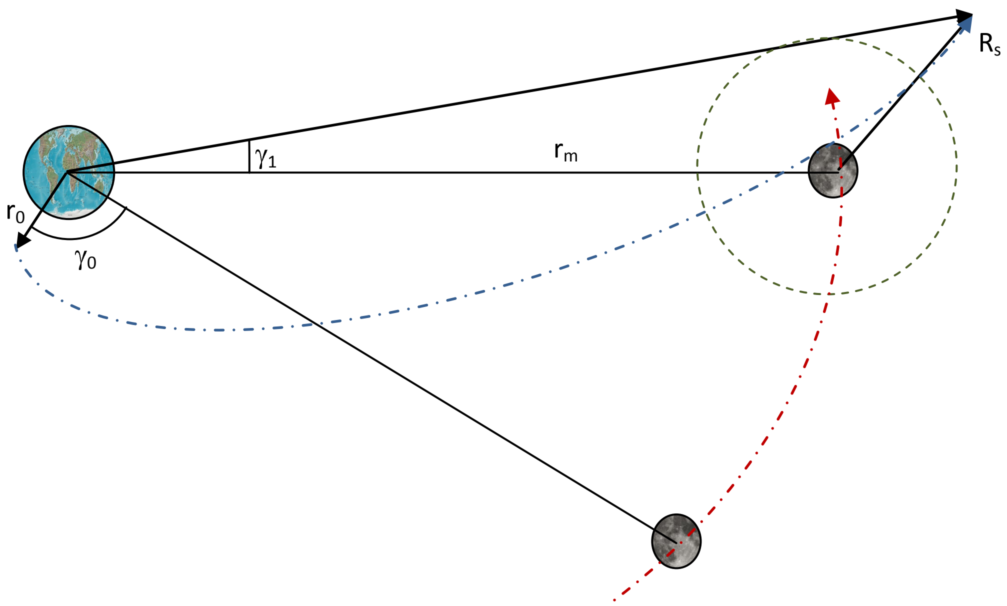
$$\boldsymbol{\gamma_{1n}} \geq \text{arctg}\left(\frac{\mathrm{r_{mn}}}{\mathrm{r_{1n}}}\right) \quad \rightarrow \quad \mathrm{t_{n+1}} = \mathrm{t_n} \quad \rightarrow \quad \gamma_{0n+1} = \gamma_0(\mathrm{t_n}) + \gamma_{1n}$$

$$\boldsymbol{\gamma_{1n}} < \text{arctg}\left(\frac{\mathrm{r_{mn}}}{\mathrm{r_{1n}}}\right) \quad \rightarrow \quad \mathrm{t_{n+1}} = \mathrm{t_n} + \Delta\mathrm{t} \quad \rightarrow \quad \gamma_{0n+1} = \gamma_0(\mathrm{t_{n+1}})$$

Cabe destacar, que el ángulo γ_1 tiene signo y por tanto si el radio final de la trayectoria está adelantado respecto al radio lunar (γ_1>0), incrementaremos el ángulo inicial para que en la siguiente iteración la posición quede retrasada ese ángulo, y si está retrasado (γ_1<0) reduciremos el ángulo para buscar el efecto contrario.

3. Nave demasiado alejada en el momento del encuentro

Otra situación derivada del error en la estimación de γ_0 es que a pesar de que γ_1 sea menor al valor máximo para una intersección el valor del radio de la nave sea tan elevado en el instante del encuentro con la Luna, que nunca llegue a situarse a una distancia menor al radio de la órbita circular.



En realidad, ésta situación es debida a la misma causa, ya que un lanzamiento con un ángulo inicial γ_0 menor, haría que la Luna se cruzase con la trayectoria antes y por tanto, el radio en el instante del encuentro fuese menor.

La forma de detectar éste error, es sencilla que solo debemos comprobar que el radio de la posición final es demasiado grande y para corregirlo reduciremos de forma iterativa el ángulo del lanzamiento hasta que lleguemos a un punto de contacto válido.

Por tanto, para corregir éste error, debemos modificar la condición anterior de la siguiente forma

$$\boldsymbol{\gamma_{1n}} \geq \operatorname{arctg}\left(\frac{r_{mn}}{r_{1n}}\right) \quad \rightarrow \quad t_{n+1} = t_n \quad \rightarrow \quad \gamma_{0n+1} = \gamma_0(t_n) + \gamma_{1n}$$

$$\boldsymbol{\gamma_{1n}} < \operatorname{arctg}\left(\frac{r_{mn}}{r_{1n}}\right)$$

$$r_1 > r_m + R_L \quad \rightarrow \quad t_{n+1} = t_n \quad \rightarrow \quad \gamma_{0n+1} = \gamma_0(t_n) + \frac{1}{2}\operatorname{arctg}\left(\frac{r_{mn}}{r_{1n}}\right)$$

$$r_1 < r_m + R_L \quad \rightarrow \quad t_{n+1} = t_n + \Delta t \quad \rightarrow \quad \gamma_{0n+1} = \gamma_0(t_{n+1})$$

6.5.2 ORBITADOR LUNAR ESMOPROP

Durante el proceso iterativo realizamos varias llamadas a un propagador de órbitas que nos permite calcular de forma precisa la posición de nuestra nave para un tiempo de vuelo determinado con unas condiciones iniciales determinadas. El propagador “esmoprop v1.0”.

El propagador “emoprop v1.0” se trata de un programa de cálculo diseñado para calcular la trayectoria lunar de una aeronave, teniendo en cuenta varios factores relevantes. Los parámetros que se deben introducir en el propagador son entre otros, los parámetros de una órbita kepleriana y la posición de la nave en el instante del lanzamiento, la fecha del lanzamiento que determinará la posición de la luna y el tiempo de vuelo. El orbitador nos devolverá la posición de la nave y su velocidad, así como la posición de la luna en varios instantes del lanzamiento, en un eje de coordenadas cartesiano con su origen n la tierra y el eje x apuntando al equinoccio de verano.

6.5.2.1 CONDICIONES INICALES DEL PROPAGADOR.

Como hemos dicho anteriormente, el propagador requiere una serie de condiciones iniciales para realizar su función. Entre otras variables que son necesarias definir, las que afectan al propósito de nuestro proyecto, pueden resumirse en dos grupos, la órbita descrita por la nave y el tiempo de vuelo.

1. **<u>Orbita descrita en el lanzamiento</u>**

Aunque los parámetros que hemos calculado, están basados en las condiciones que tiene la nave en el instante de la inyección, r_0,v_0, θ_0 y γ_0, estos parámetros corresponden a una única órbita kepleriana que puede ser definida por los parámetros orbitales clásicos descritos en la sección 5.4. Éstos son,

- **SEMI-EJE MAYOR** (a)**.**
- **ECCENTRICIDAD** (e)**.**
- **INCLINACIÓN** (i)**.**
- **LONGITUD DEL NODO ASCENDENTE** (Ω)**.**
- **ARGUMENTO DEL PERIAPSIS** (ω)

Para poder arrancar el propagador, es necesario indicarle éstos parámetros a través del archivo orbit.dat, cuyo cuerpo de texto es el siguiente.

```
//
//Orbital parameters at epoch
//

//Right Ascension
Omega = -128.646698
//Inclination
i = 25.299999
//Argument of periapsis
w=0.000000
//Mean Anomally
n = 0.000009
//Semimayor axix
a = 170549.321443
//Eccentricity
e = 0.960732
//True anomally at epoch
nu0 = 0.000000

//
//Attitude and angular velocity at epoch
//Angles in degrees, angular speeds in rad/s
//
pitch = 0.000000
roll  = 0.000000
yaw   = 0.000000
wx = 0.000000
wy = 0.000000
wz = 0.000000


//Epoch Time
year = 2010
```

month = 1

day = 21

hour = 9

minute = 0

seconds = 0

Si observamos el cuerpo del fichero, vemos que está dividid en tres partes.

En el primer grupo “orbital parameters at epoch” además de los cinco parámetros orbitales clásicos, debemos indicarle la posición actual de nuestra nave en el instante del lanzamiento. Esto lo realizamos a partir del ángulo ν, que nos indica el ángulo barrido por nuestra posición respecto del semieje mayor de la trayectoria, tomando el origen el perigeo. En nuestro caso ν_0 siempre será cero ya que una de las condiciones iniciales que proponemos es que el ángulo inicial del lanzamiento sea de cero grados. Esto únicamente sucede en el apogeo y en el perigeo por lo que en el momento de partida, nuestra nave siempre estará situada en el perigeo de la órbita.

En el siguiente grupo “Attitude and angular velocity at epoch” se indican parámetros para describir un posible movimiento local indicando los ángulos de orientación de la nave en un eje de coordenadas local (Angulos de Pitch, Yaw y roll) así como las velocidades angulares de cada uno de los ejes. En nuestro caso, simplificaremos a un movimiento estático local, dando un valor de cero a todos los parámetros.

En el último grupo, indicamos la fecha del lanzamiento. Esto es debido a que el propagador, no sólo nos da información de la posición de nuestra nave, sino también de la posición de la luna.

Los parámetros de la órbita se calculan a partir de las ecuaciones expuestas en el capítulo 5 usando los datos de partida de la forma siguiente.

Siguiendo la metodología descrita en el apartado 5.6, debemos aplicar las siguientes ecuaciones utilizando los datos de partida r_0, v_0, θ_0, y r_1

$$\varepsilon = \frac{v_0^2}{2} - \frac{\mu}{r_0}$$

$$h = R_0 . V_0 . \cos(\phi_0)$$

$$a = \frac{-\mu}{2\varepsilon}$$

$$e = \sqrt{1 - \frac{h^2}{\mu a}}$$

$$\upsilon_0 = \arccos\left(\frac{p - r_0}{r_0 e}\right)$$

$$\Omega = -\gamma_0 - \alpha_{0moon}$$

Como hemos indicado antes, el ángulo ν_0 será cero y para el caso bidimensional estableceremos el parámetro ω en 0 y la inclinación i de la trayectoria en la inclinación de la órbita lunar en el momento del vuelo.

Tiempo de vuelo

Para poder definir el tiempo de vuelo de nuestra trayectoria, es necesario modificar el archivo "integrator.dat" que define los parámetros del integrador utilizado por el orbitador. El cuerpo del archivo es el siguiente,

```
//	Total integration time in seconds
integrationtime = 313596.6907943// Parameters required by the numerical integrator odeint
//	from Numerical Recipes in C (Press, W.H.)
//	All times and delta steps in seconds
kmax = 100000
odeintacc = 1e-6
odeinthmin = 1e-3
odeinth1 = 1e-3

//	Parameter for the old and good RK4
//	All times and delta steps in seconds
dtrk4 = 1.0
```

El único parámetro, que debemos atender, es el relativo al tiempo de integración "integrationtime", donde deberemos indicar el tiempo de nuestro vuelo en segundos.

6.5.2.2 DATOS DE SALIDA DEL ORBITADOR

Los datos de salida proporcionados por el orbitador, son escritos en el archivo output_tmp.dat.

El orbitador escribe sobre éste archivo una línea de programa por cada salto temporal que realiza, generando una serie de puntos suficientes para describir la trayectoria de la nave, es decir, nos muestra una foto de la posición de la nave y de la luna en diferentes instantes del vuelo.

Cada línea de programa está estructurada en diecisiete columnas separadas por “;” con la siguiente información.

- Tiempo en segundos desde el lanzamiento (epoch). (C1)
- Coordenada r_{X1} en km (Posición final de la nave) (C2)
- Coordenada r_{Y1} en km (Posición final de la nave) (C3)
- Coordenada r_{Z1} en km (Posición final de la nave) (C4)
- V_{1X} en km/s (Velocidad final de la nave) (C5)
- V_{1Y} en km/s (Velocidad final de la nave) (C6)
- V_{1Z} en km/s (Velocidad final de la nave) (C7)
- Cuatro parámetros de Euler que describen la actitud del vehículo. (C8,9,10 y 11)
- Tres coordenadas para describir la velocidad angular del vehículo. (C12,13 y 14)
- Coordenada r_{Xm} en km (Posición de la Luna) (C15)
- Coordenada r_{Ym} en km (Posición de la Luna) (C16)
- Coordenada r_{Zm} en km (Posición de la Luna) (C17)

La estructura del fichero será por tanto,

$C1_0$; $C2_0$; $C3_0$; $C4_0$; $C5_0$; $C6_0$; $C7_0$; $C8_0$; $C9_0$; $C10_0$; $C11_0$; $C12_0$; $C13_0$; $C14_0$; $C15_0$; $C16_0$; $C17_0$;

$C1_1$; $C2_1$; $C3_1$; $C4_1$; $C5_1$; $C6_1$; $C7_1$; $C8_1$; $C9_1$; $C10_1$; $C11_1$; $C12_1$; $C13_1$; $C14_1$; $C15_1$; $C16_1$; $C17_1$;

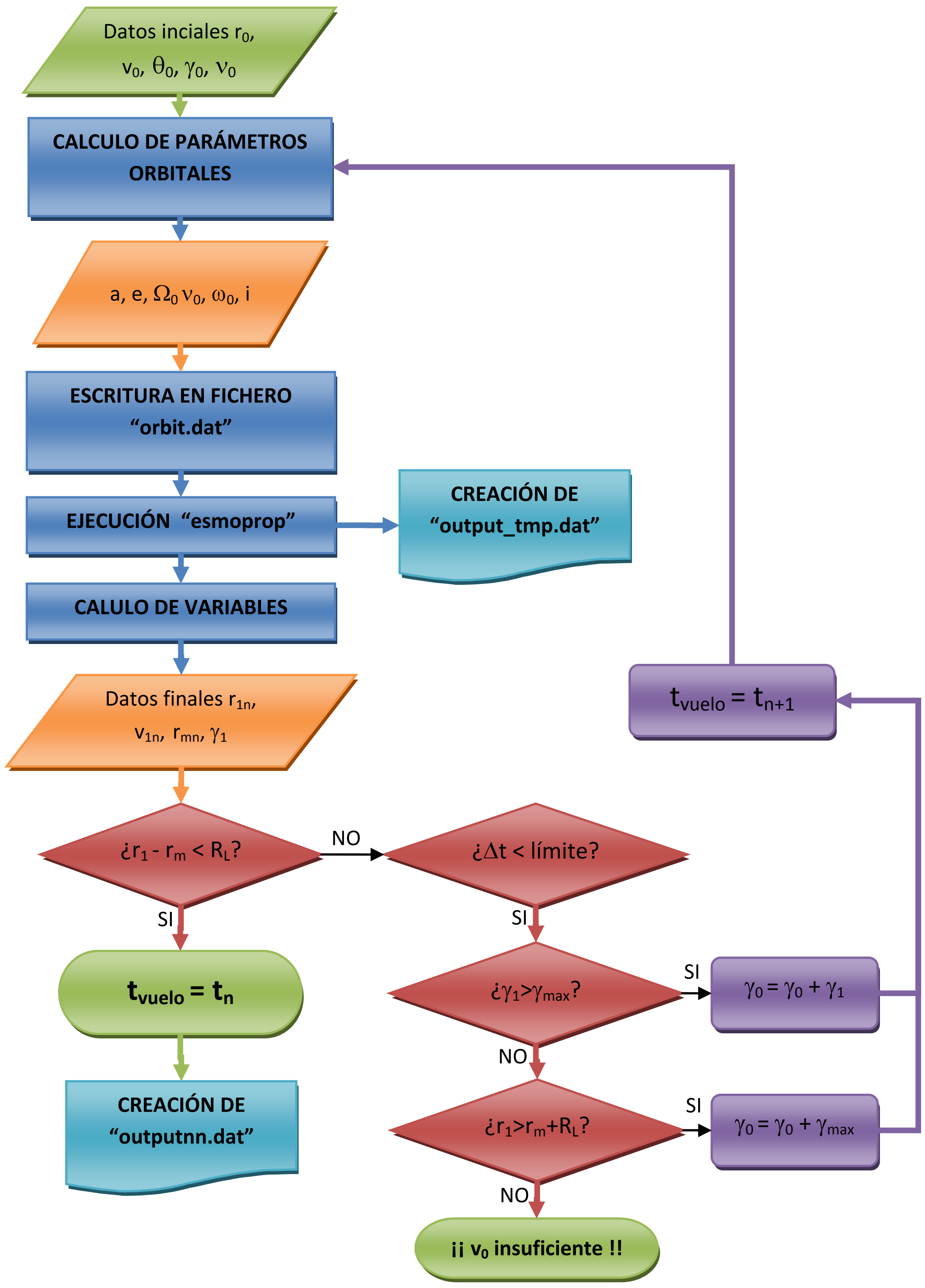
…

$C1_n$; $C2_n$; $C3_n$; $C4_n$; $C5_n$; $C6_n$; $C7_n$; $C8_n$; $C9_n$; $C10_n$; $C11_n$; $C12_n$; $C13_n$; $C14_n$; $C15_n$; $C16_n$; $C17_n$;

Para el propósito de nuestro proyecto, no tendremos en cuenta ni los parámetros de Euler ni la velocidad angular de la nave.

6.5.3 ALGORITMO DE PROGRAMA

Una vez explicadas las peculiaridades introducidas en el programa, podemos desarrollar un poco más el algoritmo de cálculo utilizado para obtener el tiempo de vuelo correcto.

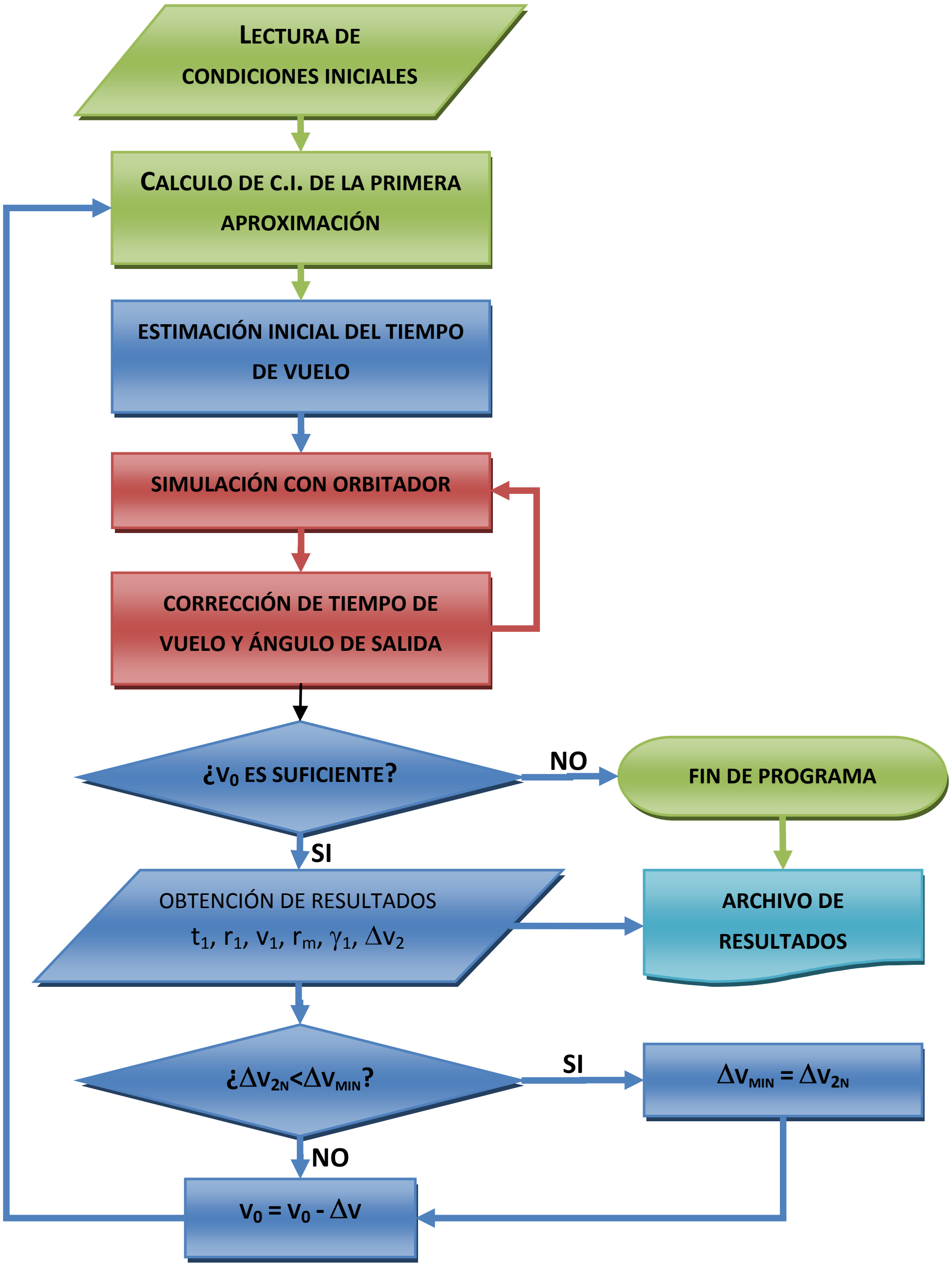


En el algoritmo, hemos incluido un paso más cuando hemos calculado un tiempo de vuelo con éxito, la creación de un archivo de salida de datos “outputn.dat” que no es más que una copia del fichero de salida del orbitador. La intención es crear un registro de ficheros de salida de cada uno de los lanzamientos, para tener una visión gráfica de cada uno de los lanzamientos a diferentes velocidades.

6.6 OPTIMIZACIÓN ENERGÉTICA, CORRECCIÓN DE V0

Hasta ahora, hemos explicado el procedimiento que seguimos para obtener el instante en el que nuestra nave se encuentra con la órbita circular que queremos describir alrededor de la Luna, sin embargo, si no realizásemos en ese punto concreto una corrección de velocidad, para poner a la nave en órbita, ésta seguiría su trayectoria y colisionaría con la Luna o se escaparía de su campo gravitatorio.

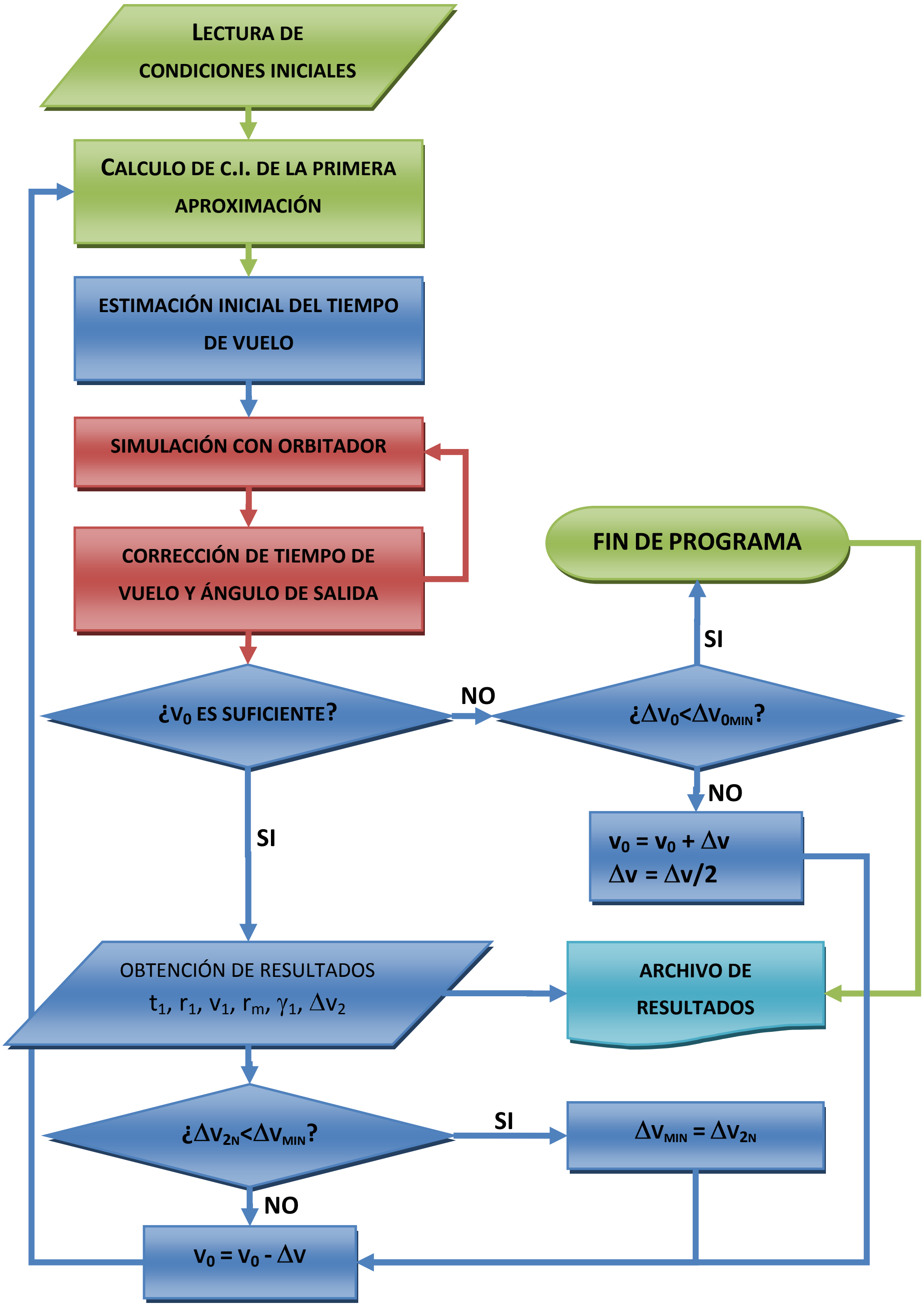
Dado que el objetivo de la misión es situar la nave en una órbita alrededor de la luna, debemos calcular la energía necesaria para modificar la velocidad de la nave, o lo que es lo mismo, el incremento de velocidad necesario. Una vez calculado, realizaremos varios lanzamientos hasta que la velocidad inicial no sea suficiente para después escoger el lanzamiento que requiera un menor incremento de velocidad. Para perseguir éste objetivo, asociado a la rutina anteriormente descrita, seguimos el siguiente esquema.



Por otro lado, dado que la velocidad inicial del lanzamiento que toca la órbita circular que queremos describir alrededor de la Luna disminuye con la longitud del semieje mayor de la trayectoria, si encontramos la velocidad mínima suficiente que consigue contactar con la órbita circular en su apogeo tendremos probablemente uno de los puntos más óptimos desde el punto de vista energético, ya que la impulsión inicial del lanzamiento será menor. Para poder calcular, éste punto debemos introducir un proceso iterativo que nos permita aproximar el lanzamiento a éste punto óptimo.

De manera similar al proceso de cálculo del tiempo de vuelo, si determinamos que la velocidad del lanzamiento no es suficiente, daremos un paso atrás y reduciremos la velocidad con la mitad de incremento.

Una vez añadido éste proceso, el diagrama quedará de la siguiente forma.



6.6.1 CALCULO DEL IMPULSO Y EL PROCESO ITERATIVO

Para el cálculo del impulso de velocidad que pondrá en órbita a nuestra nave, si tenemos en cuenta el modelo bidimensional, deberemos seguir el siguiente procedimiento.

Conocidos v_{1x}, v_{1y} , r_{1x}, r_{1y}, r_{mx} y r_{my}.

$$\overrightarrow{r_2} = \overrightarrow{r_1} - \overrightarrow{r_m}$$

Los vectores unitarios perpendiculares a r_2 y a r_m, definen las direcciones de la velocidad de la órbita lunar necesaria y la velocidad de la luna respecto a la tierra respectivamente. Estos vectores, pueden calcularse para el caso bidimensional como,

$$\vec{u}_{vmoon} = \frac{-r_{my}}{\sqrt{r_{mx}^2 + r_{my}^2}}\vec{i} + \frac{r_{mx}}{\sqrt{r_{mx}^2 + r_{my}^2}}\vec{j}$$

$$\vec{u}_{vcs2} = \frac{-r_{2y}}{\sqrt{r_{2x}^2 + r_{my}^2}}\vec{i} + \frac{r_{2x}}{\sqrt{r_{2x}^2 + r_{2y}^2}}\vec{j}$$

Las velocidades serán por tanto,

$$\vec{v}_m = |\vec{v}_m|.\vec{u}_{vmoon}$$

$$\vec{v}_{cs2} = |\vec{v}_{cs2}|.\vec{u}_{vcs2}$$

Donde tomaremos el módulo de v_m como constante e igual a 1,018 km/sg y el módulo de v_{cs2} como,

$$V_{CS2} = \sqrt{\frac{\mu_m}{R_L}}$$

Una vez calculados los vectores, podemos calcular el incremento de velocidad como

$$\overrightarrow{v_2} = \overrightarrow{v_1} - \overrightarrow{v_m}$$

$$\Delta\overrightarrow{v_2} = \overrightarrow{v_{cs2}} - \overrightarrow{v_2}$$

6.7 AMPLIACIÓN AL MODELO 3D

Hasta el momento, sólo hemos desarrollado un programa, basado en un modelo bidimensional suponiendo que las órbitas de la trayectoria y de la Luna, son coplanarias con el plano ecuatorial, tomando como válidas las proyecciones de los resultados del propagador sobre el plano ecuatorial.

Esto, es erróneo por varias razones. En primer lugar, la proyección de una órbita circular sobre el plano ecuatorial, no sería un círculo, sino que sería una elipse, lo cual nos lleva a que aun en el caso de que ambas trayectorias se encuentren en el mismo plano, es muy probable que el impulso lo demos a una distancia equivocada y el segundo lugar, aunque la inclinación de ambas órbitas sea la misma, sólo serán lanzamientos coplanarios si también coincide su línea de nodos.

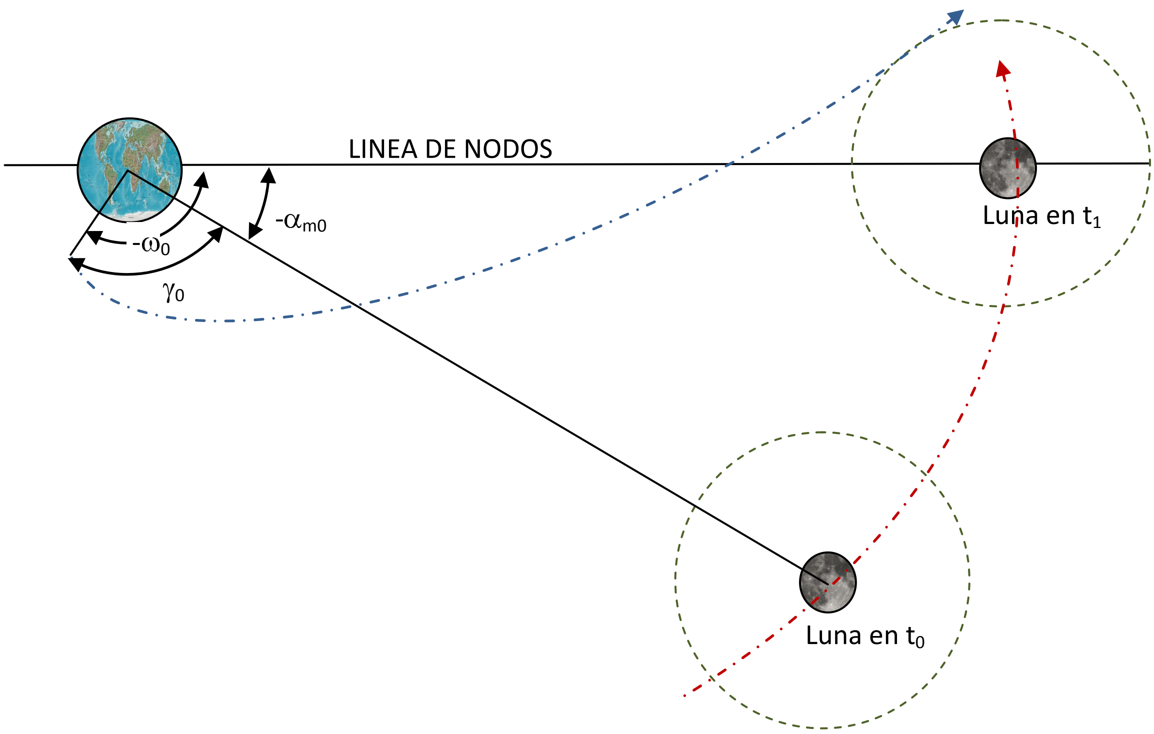
6.7.1 LANZAMIENTO COPLANARIO TRIDIMENSIONAL

Ambas situaciones, pueden corregirse de diferentes maneras. La más sencilla, sería escoger una órbita coplanaria con la luna haciendo coincidir tanto la inclinación i como el ángulo Ω de nuestra trayectoria sobre la línea de nodos. Si hacemos esto, debemos tener en cuenta que para varia la orientación de nuestra trayectoria en base a las estimaciones de γ_0, deberemos variar el ángulo ω y no Ω.

Si hacemos esto, conocidos Ω_m y ω_m en el momento del lanzamiento, hacemos que nuestra órbita sea,

$$\Omega_{trayectoria} = \Omega_m \qquad\qquad i_{trayectoria} = i_m$$

Para poder calcular la corrección que debemos hacer sobre la órbita, debemos tener en cuenta las siguientes relaciones. Si hacemos coincidir la línea de nodos de ambas trayectorias, en el plano de la trayectoria tendremos lo siguiente.

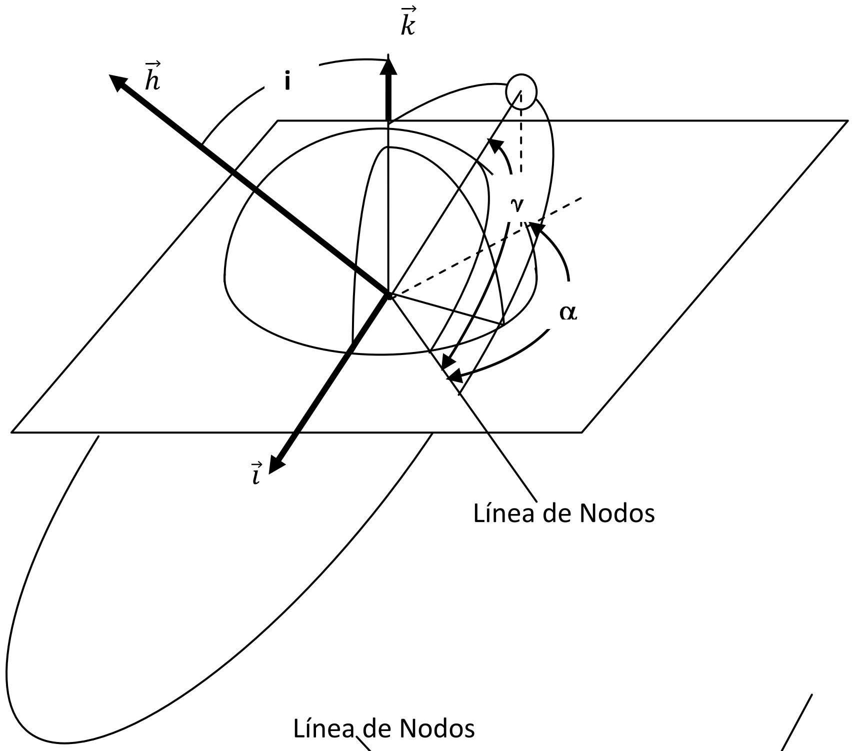


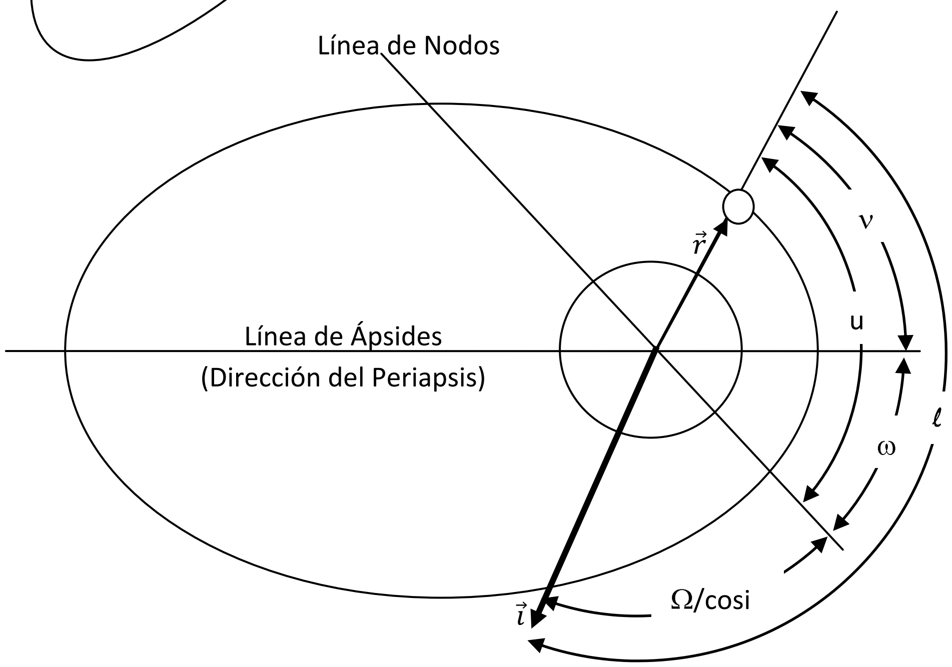
$$-\omega_{0\,orbita} = -\alpha_{m0} + \gamma_0$$

$$\omega_{0\,orbita} = \alpha_{m0} - \gamma_0$$

Sin embargo, en el programa, nosotros calculamos γ_0 en base a las proyecciones sobre el plano ecuatorial, y por tanto, lo que calculamos realmente es su proyección α así que para que la relación anterior sea del todo correcta, debemos corregir la ecuación.

$$\omega_{0\ orbita} = \alpha_m - \frac{\gamma_0}{\cos(i)}$$





Aunque éste lanzamiento sí es coplanario, para obtener una solución exacta del todo, necesitamos modificar el programa ya que la iteración que realizamos, es en base a la proyección del plano de las órbitas sobre el plano ecuatorial. En éste plano, una órbita circular, se proyectaría como una elipse, y por tanto, el radio de contacto buscado, no sería el correcto.

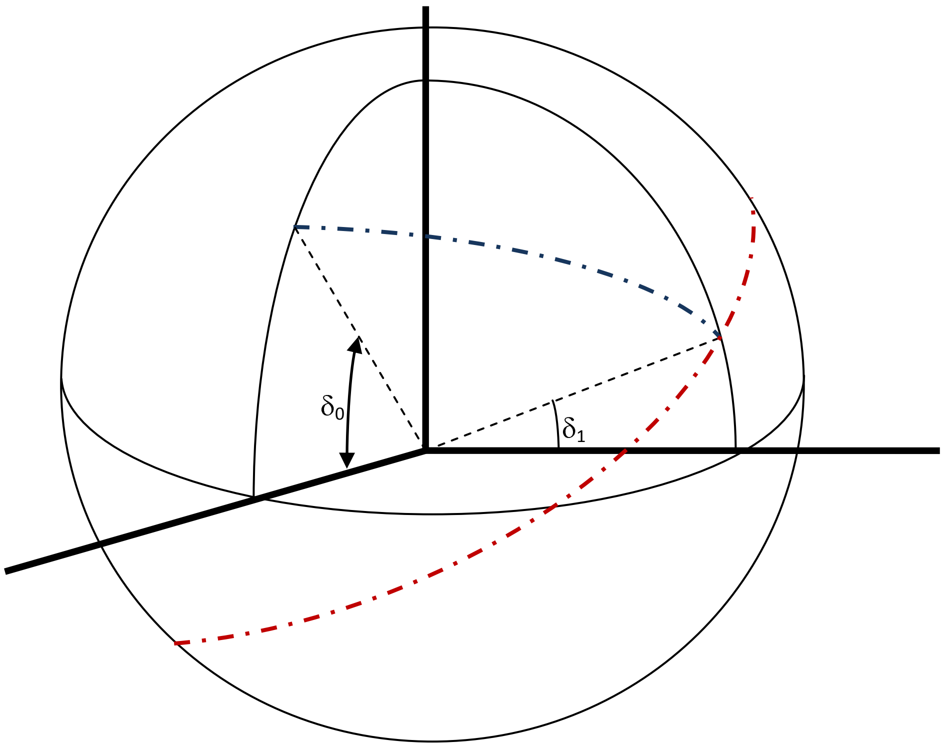
La forma más sencilla de resolver éste problema, es calcular los radios utilizados para las iteraciones, a partir de sus tres coordenadas.

6.7.2 LANZAMIENTO NO COPLANARIO

Dado que es posible que describir una órbita concreta no sea posible en todos los casos, es recomendable buscar una posible solución a partir de un lanzamiento no-coplanario.

Para que dos cuerpos describiendo trayectorias en planos distintos lleguen a encontrarse, debemos obligar a que el momento del encuentro se realice en la recta intersección de ambos planos.

Si nos fijamos en la figura siguiente, podemos entender mejor la condición.



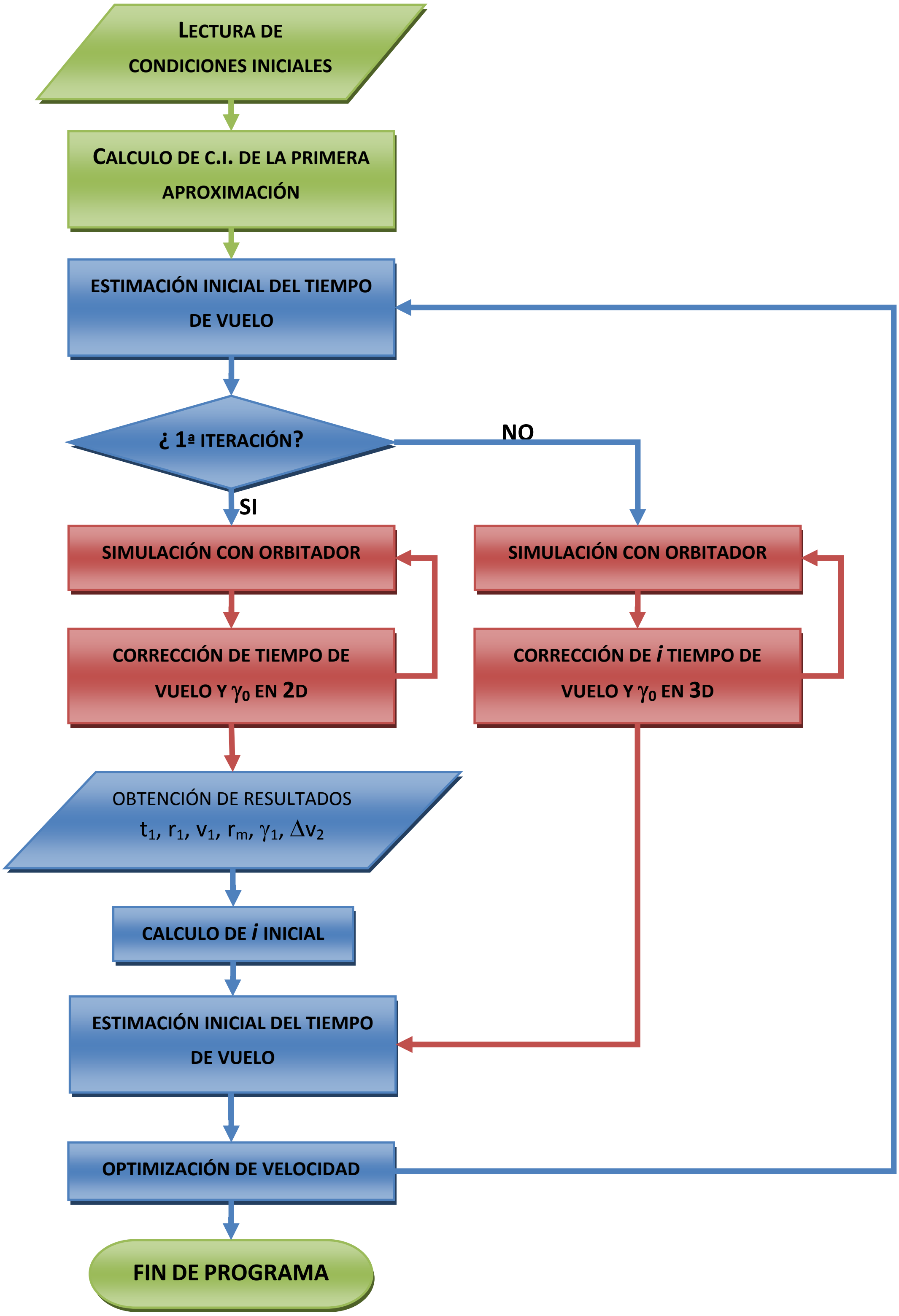
Para poder cumplir ésta condición, debemos asegurarnos, que la inclinación de la órbita que deseamos describir, sea la correcta.

Si nos fijamos en la figura, la inclinación del plano de la trayectoria, viene determinada por los ángulos δ_0 y δ_1. En el caso de nuestro lanzamiento, si tomamos como condiciones iniciales ω=0 y ν_0=0, nuestra δ_0 será también 0 ya que nuestra nave partirá desde la línea de ápsides (ν_0=0) que estará situada en la línea de nodos (ω=0), es decir, que partiremos del plano ecuatorial.

Por otro lado, el ángulo δ_1, debe coincidir con la posición final de la luna. Aunque nosotros en realidad, lo que buscamos es situarnos a una distancia R_L de la luna, haciendo coincidir la línea de corte de los planos de las trayectorias con el vector posición de la Luna, realizaremos un corte a la esfera de posibles trayectorias lunares con lo que con un impulso de velocidad adecuado, describiremos la trayectoria deseada.

Dado que la inclinación de nuestra trayectoria viene determinada por el producto vectorial de ambos vectores (posición inicial de la nave y posición final de la Luna), debemos calcularla después de haber realizado una primera estimación del tiempo de vuelo.

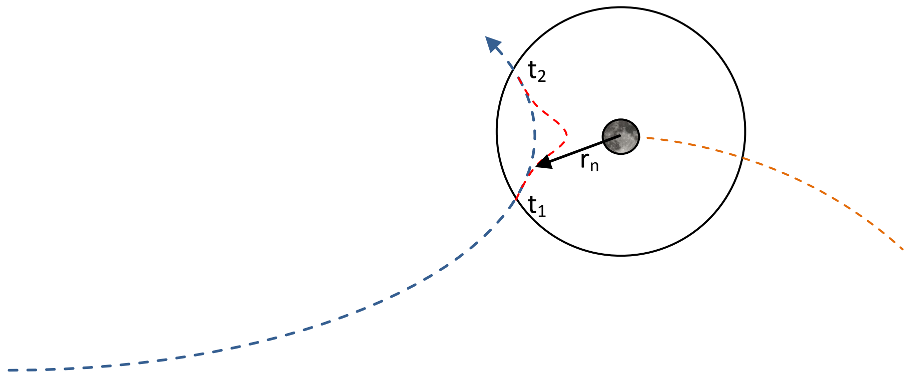
El procedimiento a seguir para la resolución de éste método sería por tanto el siguiente.


LECTURA DE CONDICIONES INICIALES
CALCULO DE C.I. DE LA PRIMERA APROXIMACIÓN
ESTIMACIÓN INICIAL DEL TIEMPO DE VUELO
¿ 1ª ITERACIÓN?
NO
SI
SIMULACIÓN CON ORBITADOR
CORRECCIÓN DE TIEMPO DE VUELO Y γ_0 EN 2D
SIMULACIÓN CON ORBITADOR
CORRECCIÓN DE *i* TIEMPO DE VUELO Y γ_0 EN 3D
OBTENCIÓN DE RESULTADOS
$t_1, r_1, v_1, r_m, \gamma_1, \Delta v_2$
CALCULO DE *i* INICIAL
ESTIMACIÓN INICIAL DEL TIEMPO DE VUELO
OPTIMIZACIÓN DE VELOCIDAD
FIN DE PROGRAMA

6.7.3 CORRECCIÓN DEL MÉTODO ITERATIVO PARA EL CASO TRIDIMENSIONAL

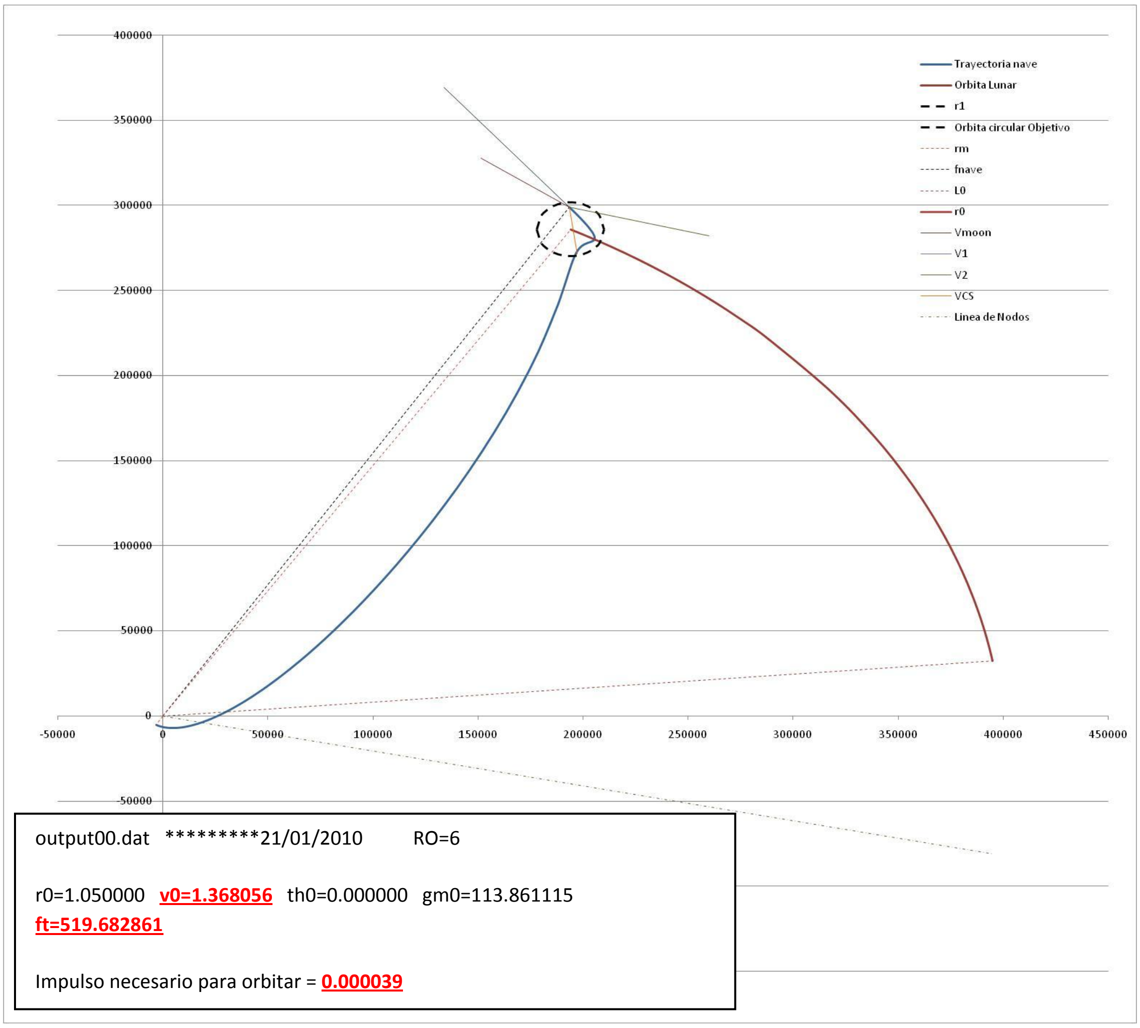
Aunque a priori, el método iterativo puede ser perfectamente válido para el caso tridimensional, existe una pequeña introducción que es necesario introducir para optimizar el proceso.

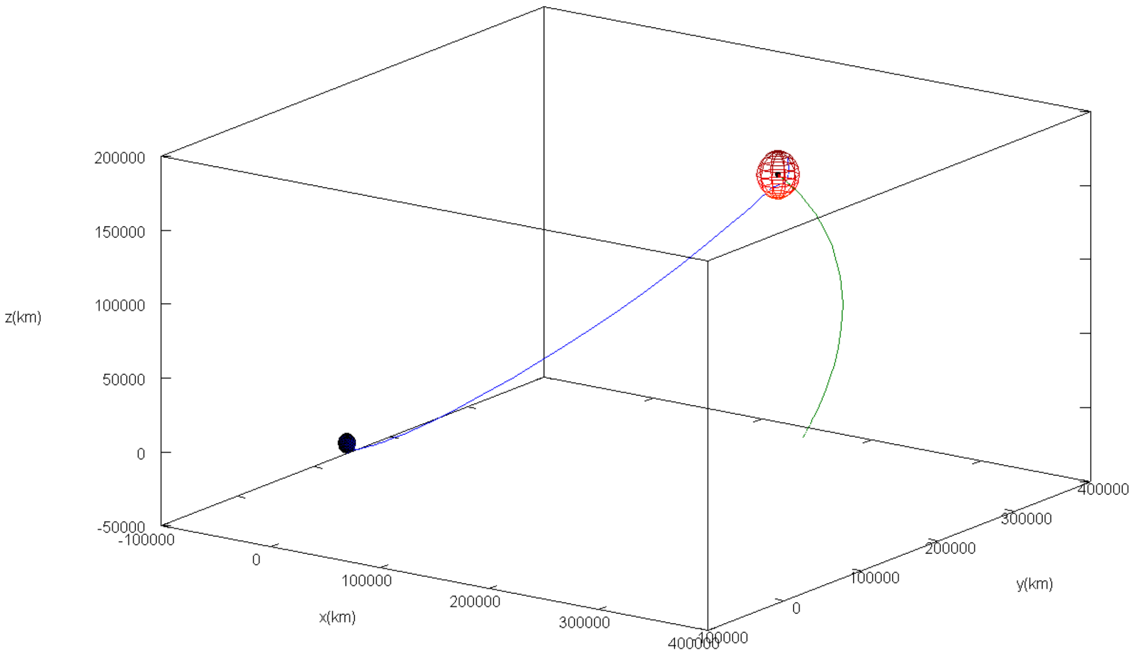
Dado que la primera aproximación está basada en un modelo simplificado, en el cual no se tiene en cuenta la atracción de la luna, es muy probable que una vez ejecutado el lanzamiento bajo un modelo más complejo, la posición real de la nave para el mismo tiempo de vuelo sea más cercana a la luna de lo que la simplificación estimaba en un principio. A una distancia lo suficientemente cercana, la trayectoria de la nave, puede verse muy afectada por la atracción luna, presentando un máximo local en las proximidades de la luna.

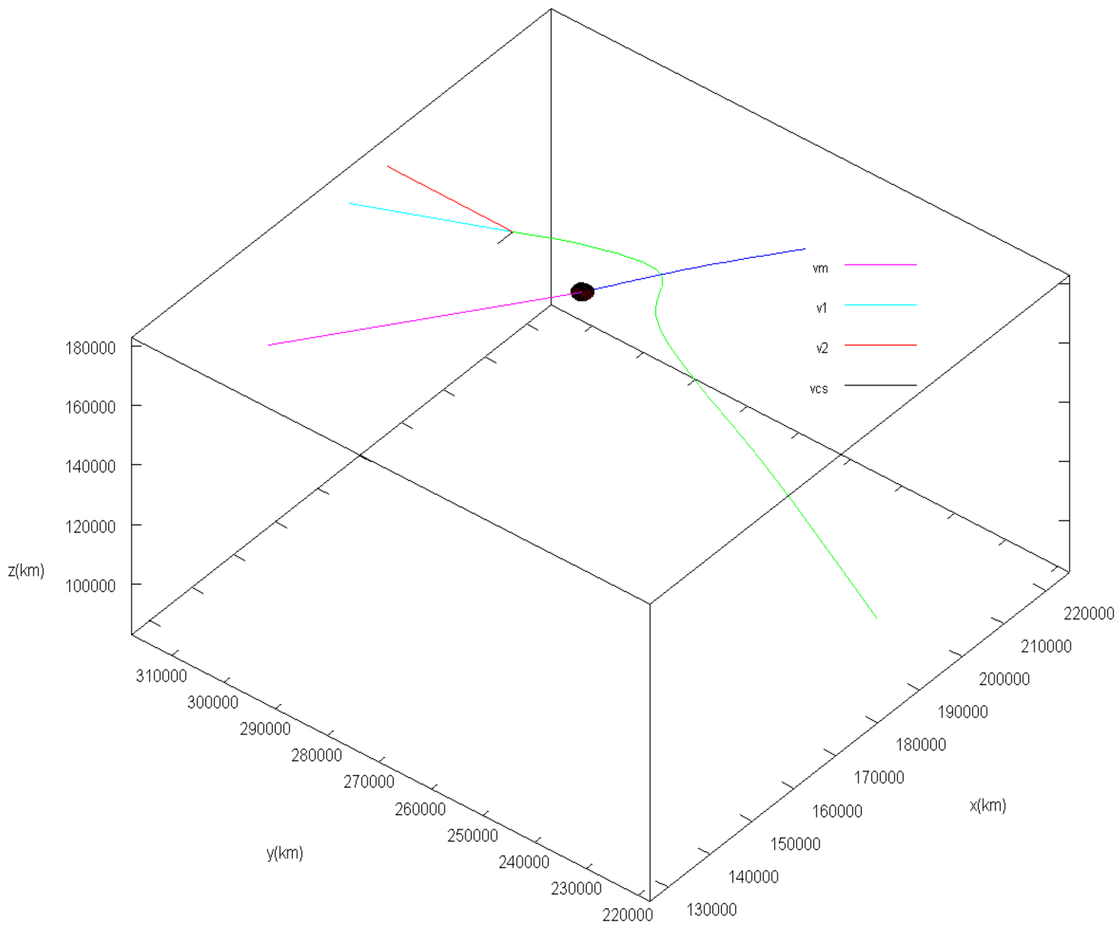


Aunque en principio, la iteración nos llevará a un resultado válido, si observamos el diagrama del capítulo 6.5, no sólo analizamos el valor absoluto de la distancia a la luna, sino que lo que determina el punto de convergencia del método y por tanto la evolución del tiempo de vuelo en el método, no es la distancia a la luna, sino el crecimiento de ésta. Debido a esto, en un comportamiento como el mostrado en la figura anterior, si el punto inicial se encuentra en la posición marcada por r_n, dado que la distancia decrece al incrementar el tiempo, el método continuaría incrementando el valor de la velocidad hasta situarse probablemente en el punto de encuentro de t_2.

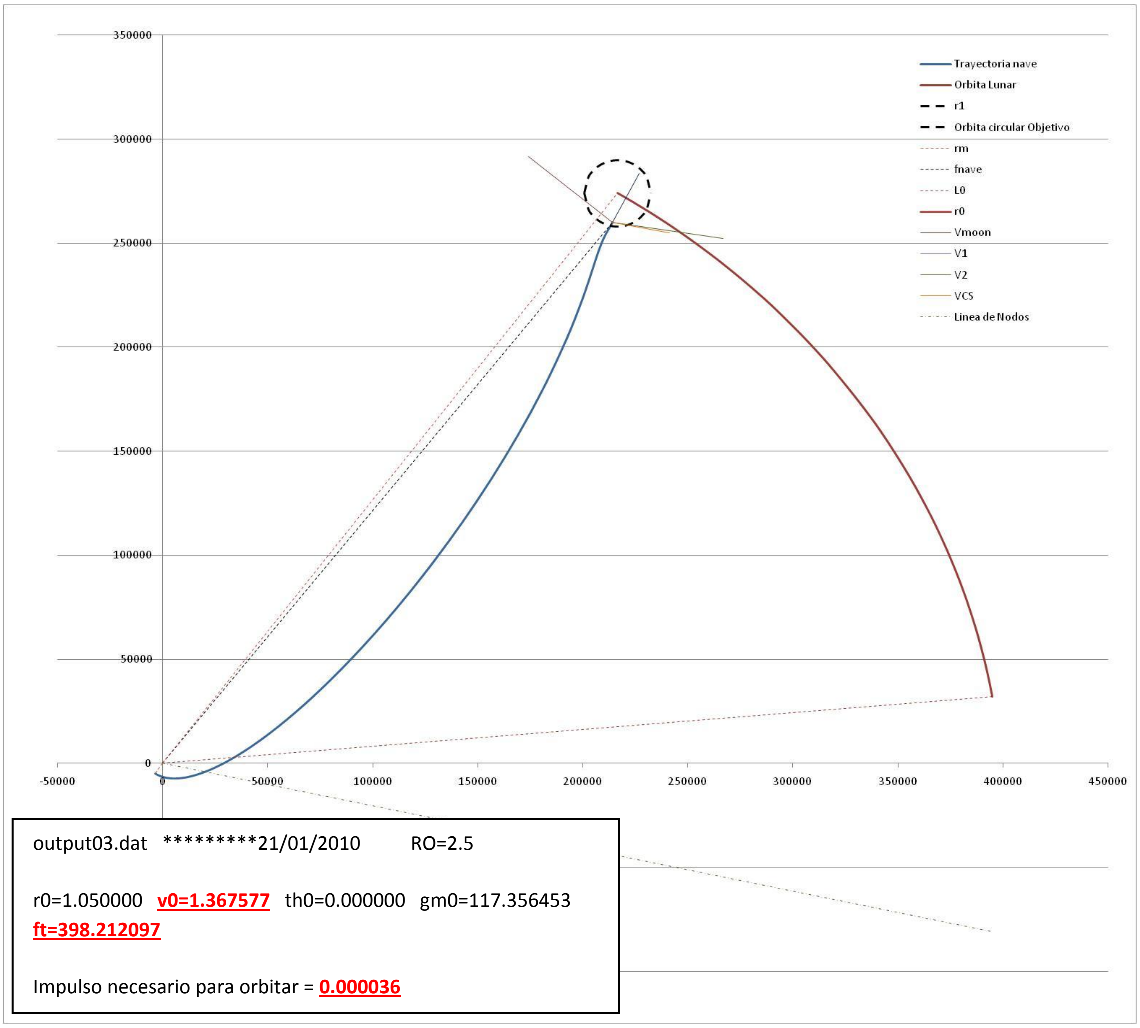
Para evitar ésta situación, debemos introducir una corrección que compruebe que siempre que el valor de la distancia sea menor a R_L, obligue decrementar el tiempo de vuelo estimado. A continuación, se muestran algunos lanzamientos realizados sin dicha corrección, y el resultado obtenido tras aplicarla.

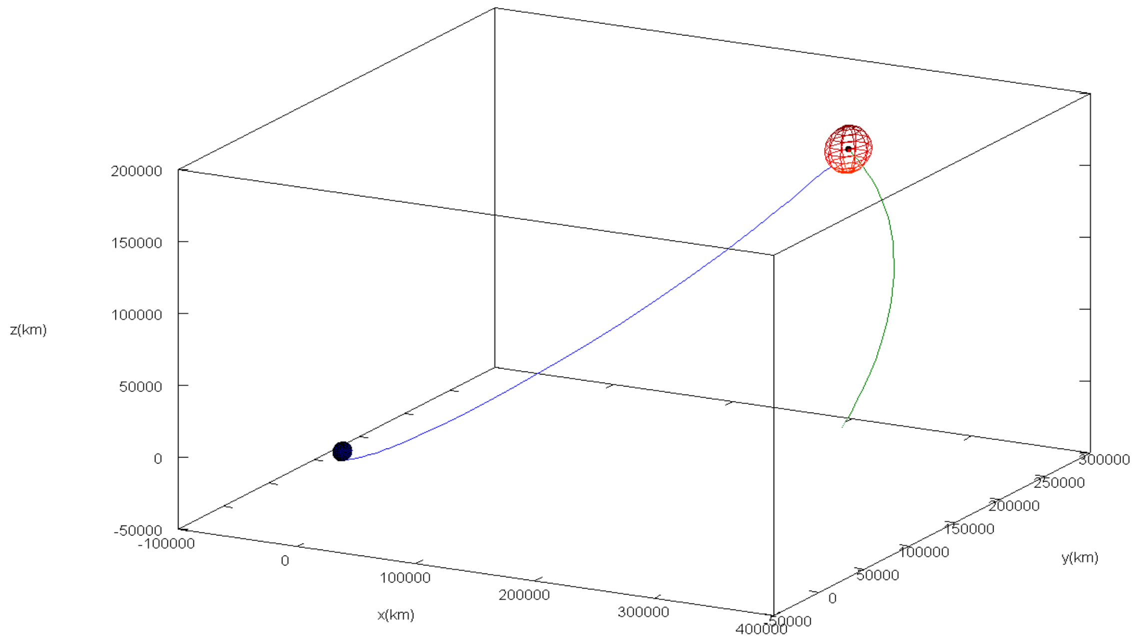


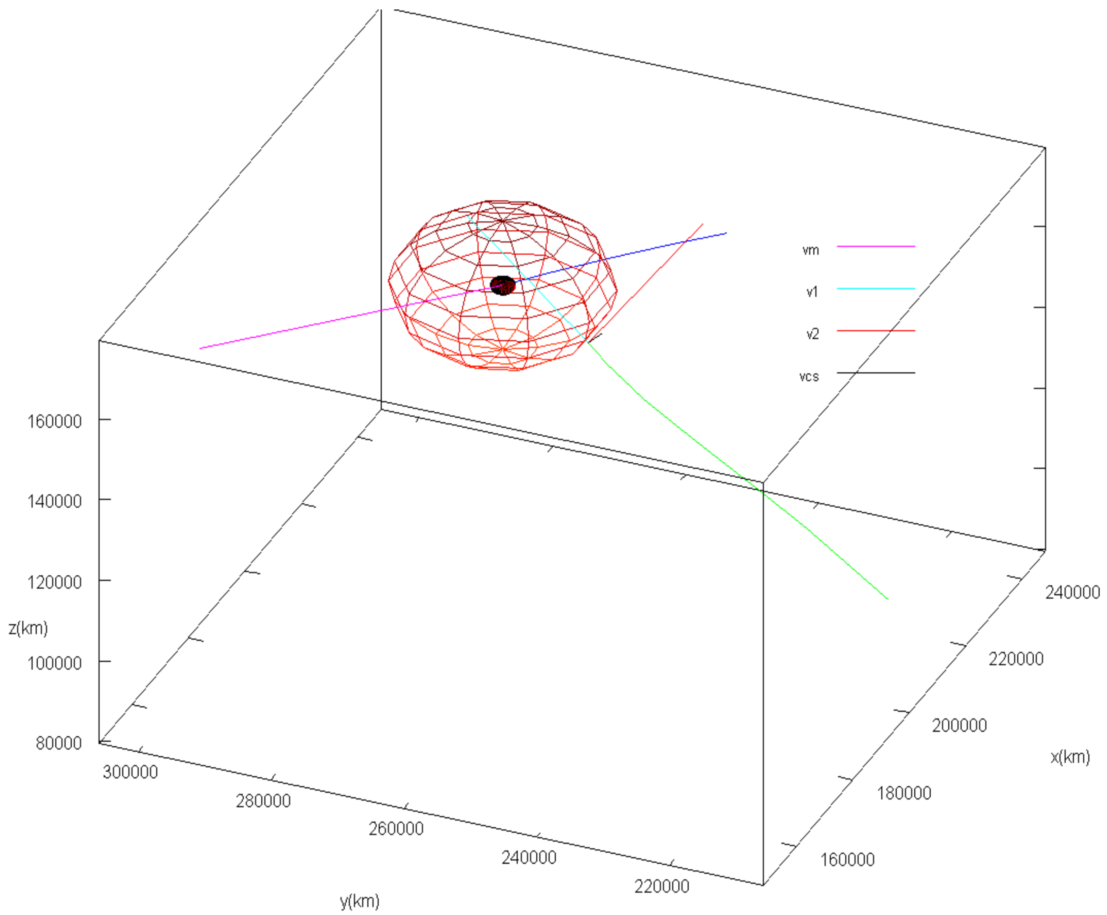

z(km)
x(km)
y(km)


vm
v1
v2
vcs
z(km)
y(km)
x(km)

Como podemos observar en las figuras, el resultado del lanzamiento, aunque se realiza a una distancia válida donde se puede realizar la maniobra para entrar en órbita con la Luna, obliga a incrementar innecesariamente el tiempo de vuelo ya que podría haberse realizado la maniobra en un instante anterior, acortando el tiempo de vuelo y optimizando por tanto la velocidad inicial del lanzamiento.





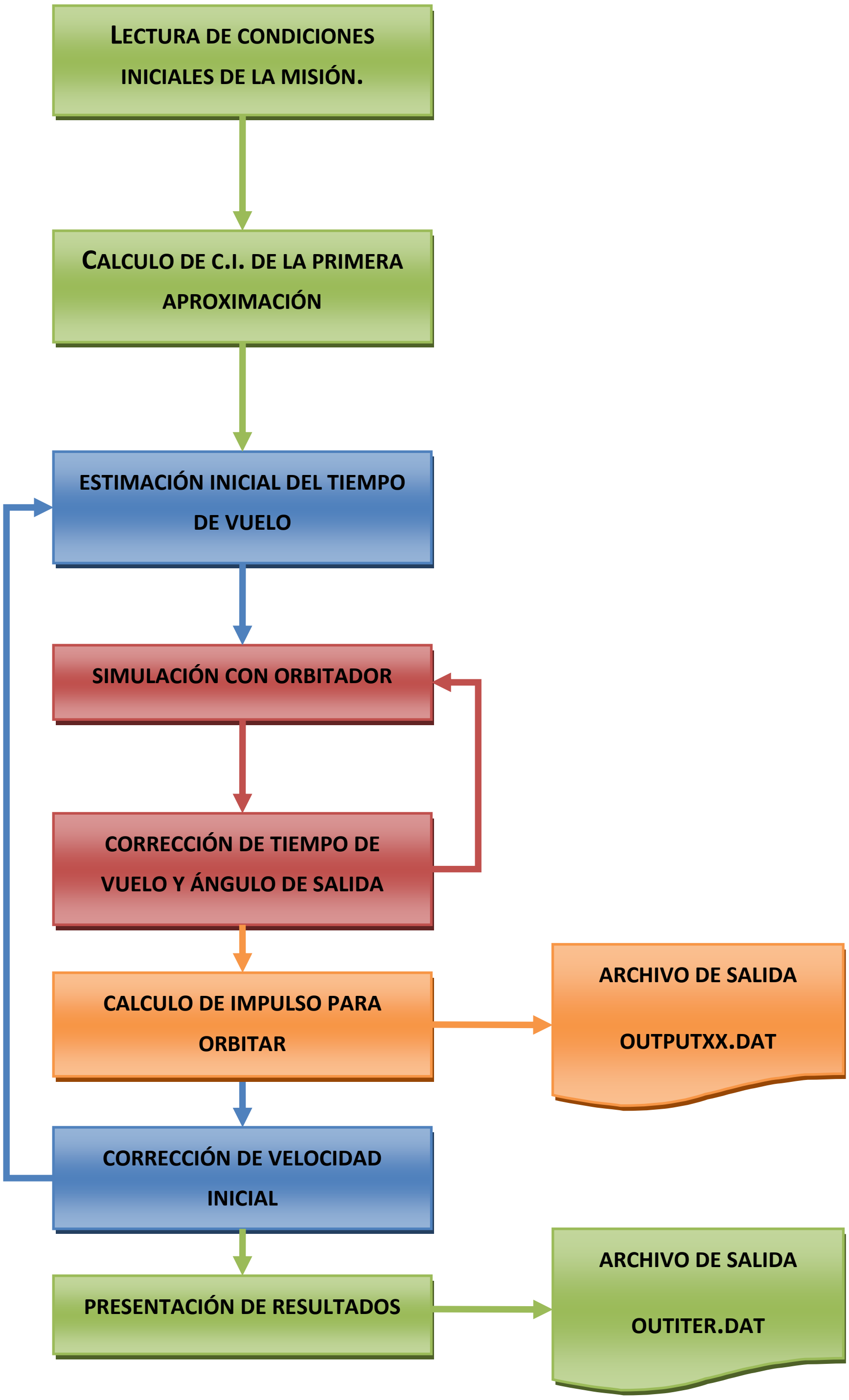


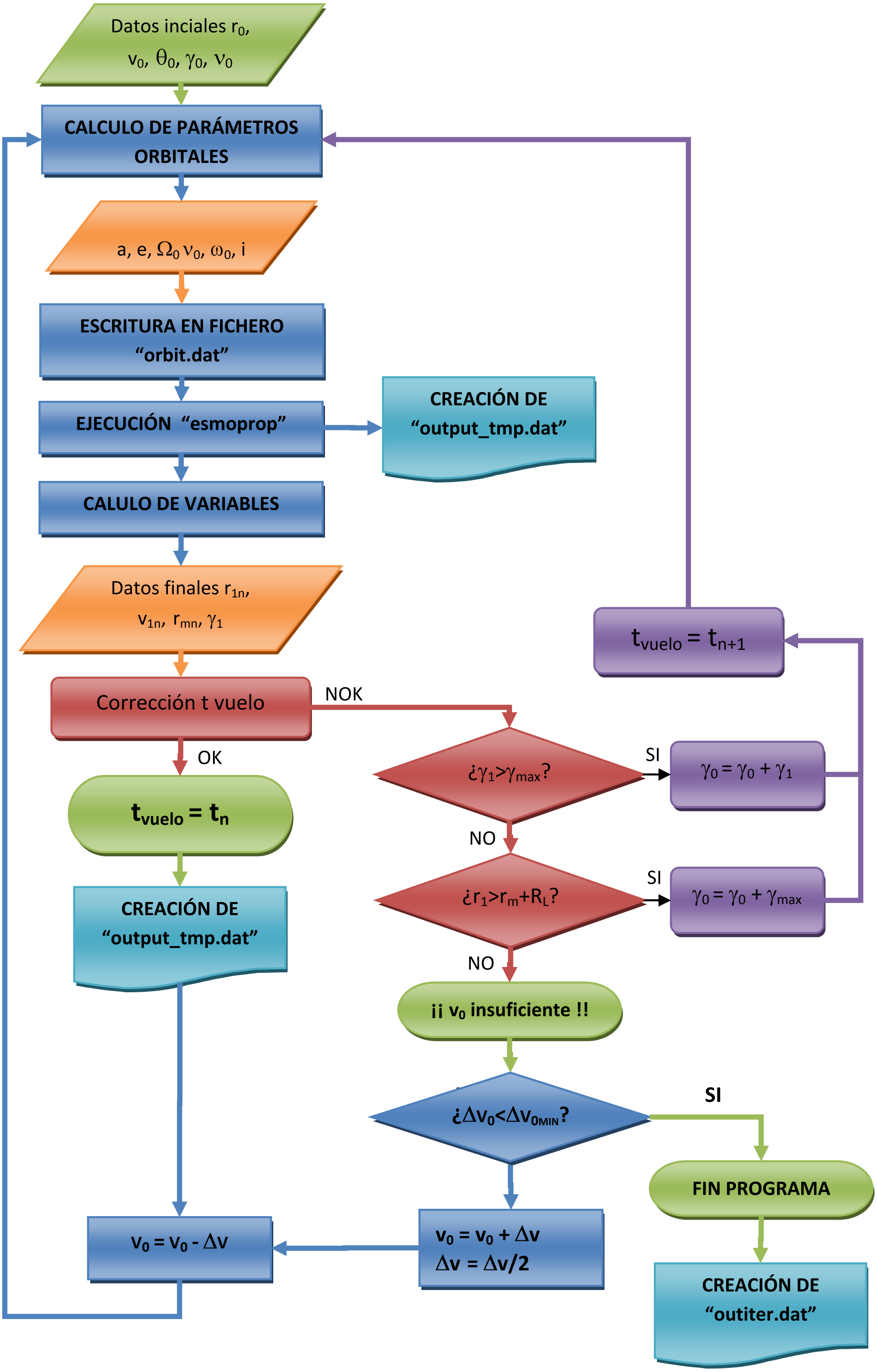
El mismo lanzamiento con la corrección introducida, reduce el tiempo de vuelo, la velocidad inicial y el incremento necesario.

7 PRESENTACIÓN DEL CÓDIGO

7.1 DIAGRAMAS DE FLUJO DEL PROGRAMA

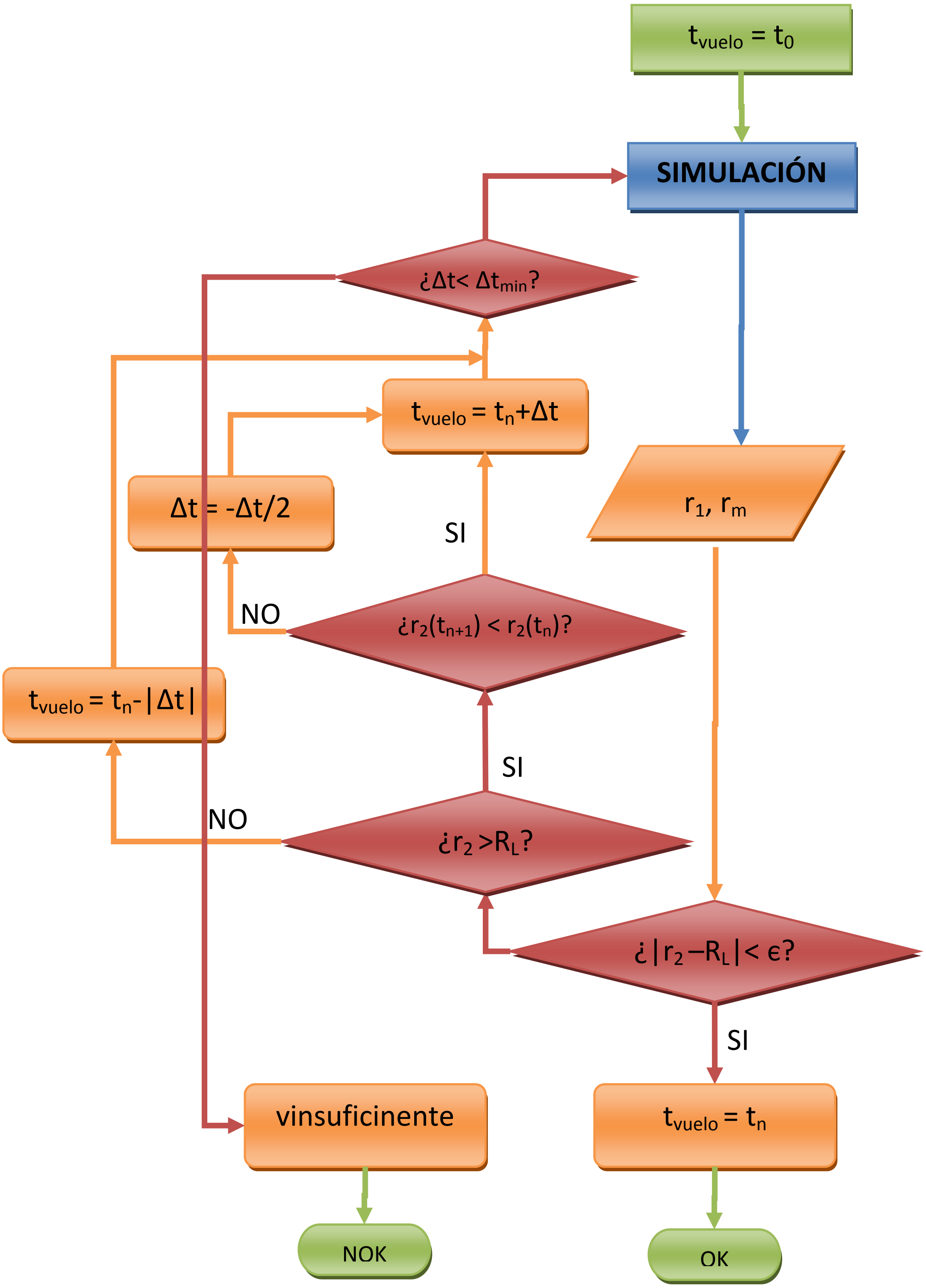
Aunque en el capítulo 6, se han expuesto todos los diagramas de flujo durante la explicación teórica sobre la que está basado todo el algoritmo. A continuación, se presenta a modo de resumen el diagrama general lógico que sigue el programa y un esquema más desarrollado.





Datos inciales r_0, v_0, θ_0, γ_0, ν_0
CALCULO DE PARÁMETROS ORBITALES
a, e, Ω_0 ν_0, ω_0, i
ESCRITURA EN FICHERO "orbit.dat"
EJECUCIÓN "esmoprop"
CREACIÓN DE "output_tmp.dat"
CALULO DE VARIABLES
Datos finales r_{1n}, v_{1n}, r_{mn}, γ_1
$t_{vuelo} = t_{n+1}$
Corrección t vuelo
NOK
OK
$t_{vuelo} = t_n$
¿γ_1>γ_{max}?
SI
$\gamma_0 = \gamma_0 + \gamma_1$
NO
¿r_1>r_m+R_L?
SI
$\gamma_0 = \gamma_0 + \gamma_{max}$
NO
CREACIÓN DE "output_tmp.dat"
¡¡ v_0 insuficiente !!
¿Δv_0<Δv_{0MIN}?
SI
FIN PROGRAMA
$v_0 = v_0 - \Delta v$
$v_0 = v_0 + \Delta v$
$\Delta v = \Delta v/2$
CREACIÓN DE "outiter.dat"

7.1.1 CORRECCIÓN DEL TIEMPO DE VUELO



7.2 CÓDIGO DEL PROGRAMA

7.2.1 ARCHIVOS INICIALES

Adjuntos al código del programa, existen algunos ficheros adicionales que son necesarios para el funcionamiento de código.

- *Init_condition.ini* Fichero donde se indican las condiciones iniciales que condicionan el lanzamiento. En él se debe indicar la fecha del lanzamiento, los parámetros de la órbita estacionaria de la nave antes de la impulsión, la precisión con la que se desea la aproximación a la órbita y el radio de la órbita circular estacionaria.
- *CVIOLclass.h* En éste fichero están las definiciones de las estructuras y objetos que se usan para definir el programa.
- *Constant.h* Archivo de definición de constantes físicas.

7.2.1.1 ARCHIVO DE CONCIDICIONES INICIALES "init_condition.ini"

```
//Condiciones iniciales de la nave
Init Position = 1.050000
Init Speed Angle = 0
//Fecha del lanzamiento
year = 2010
month = 1
day = 21
hour = 9
minute = 0
second = 0
Omega Moon orbit = 4.59 //Condiciones de la órbita Lunar en el instante del lanzamiento
//Datos de la órbita objetivo
Target Radio Moon Orbit = 6
Accuracy = 0.001
```

7.2.1.2 ARCHIVO DE DECLARACIÓN DE OBJETOS "COVIOLclass.h".

```
//Definimos la clase ship_state que definirá la situación de velocidad y posición de la nave
class init_par {public: float r0,th0,omega,rfmo,acc;int year,month,day,hour,minute,second;};
//Init radio, Init Speed, Init fly_path Angle, Init Gamma (moon position), radio of final moon
orbit, accuracy
class speed {public: float v,th;}; //Modulo de la velocidad y argumento
class ship_state {public: float re,v,th,rm,vm,ep,gm,ld,t;};
/*  re – Posición radial respect de la tierra,
    v  - Velocidad respect de la tierra,
    th – Angulo de salida respect de la tierra,
    rm – Radio desde la Luna
    vm – Velocidad desde la Luna
    ep - epsilon (Angulo complementario de theta)
    gm - gamma (Ángulo entre la línea de unión tierra-luna y la posición de la nave)
    ld - lambda (angle between line from earth to moon and radio to moon)
    t -  Tiempo de vuelo */
class orbit_data {public: float energy,momentum,eccentricity,directrix,parameter;};//
/*      e = eccentricity,
        a = directrix,
        p = semi-latus rectum o parameter */
class outdata {public: float r1,r2,rm,ang1,an2,angm,v1,boost,t,vf[3],vm[3],vcs[3];};
/*   r1 = Posición final respecto de la tierra
    r2 = Posiión final despecto de la luna
    rm = Posicion de la luna
    ang1 =
    an2 =
    v1 = Módulo de la velocidad final
    boost = Módulo del impulso para orbitar
    t = Tiempo de vuelo
    vf[3] = Vector de velocidad final
    vm[3] = Vector de velocidad de la luna
    vcs[3] = Vector de velocidad de la nave en órbita circular alrededor de la Luna
*/
```

7.2.1.3 ARCHIVO DE CONSTANTES FÍSICAS "constant.h"

```
//Definición de constantes físicas para resolver el problema
const float D = 384400;  //Distancia de la tierra a la Luna en Km
const float RS = 66300; // Radio de acción de la Luna en km
const float MU = 1; //Constante de gravitación en DU^3/TU^2
const float G = 0.89; //Constante de gravitación G de la tierra
const float WM = 2.14e-03; //velocidad angular de la Luna en rad/TU
const float DEARTH      = 6378.145; //radio de la tierra en Km (1 DU) es DU
const float TU = 0.224114333; //Equivalente en horas de Unidad de tiempo
const float MU_M = 0.01229298;//Constante gravitatoria de la Luna en DU^3/TU^2
const float VM = 0.128773211; // Velocidad de la Luna en DU/TU
const float PI = 3.14159265;
```

7.2.2 DECLARACIONES DE FUNCIONES

```
/*COVIOLC (Calculo de la Optima Velocidad Inicial para una Orbita Lunar Circular)
Éste programa calcula la velocidad necesaria*/
#include <cstdlib>
#include <iostream>
#include <math.h>
#include <string>
#include "constant.h"
#include "COVIOLCclass.h"

//Declaración de funciones
init_par data_input(); //Recoge los datos de entrada
float readfile (int,int,char*); //Lee datos de ficheros
ship_state fmosm (init_par); //Final Moon Orbit Speed Mathematics.Calcula la velocidad de la orbita circular estacionaria
float faissm (init_par); //First aproximation Init ship speed Mathematics, Calcula la velocidad de la orbita circular estacionaria
ship_state faftm (init_par,float); //First aproximation Flight time and ship state mathematics, Calcula una primera aproximación del tiempo de vuelo
outdata ssitm (ship_state,float,init_par); //Ship State in time mathematics, Ejecuta el propagador para calcular el estado de la nave tras un timepo
int rename_file(char*); //Copia un fichero con otro nombre
```

7.2.3 PROGRAMA PRINCIPAL

```
int main(int argc, char *argv[])
{
//Definición de variables
init_par ic; //Vector de condiciones iniciales
ship_state sh0; //Estado inicial de la nave
ship_state shf,sh0opt; //Estado final de la nave y estado inical para lanzaminto óptimo
outdata outn[2]; //Estado final de la nave y estado final iterativo_n
ship_state shm; //estado de la nave en órbita estacionaria alrededor de la Luna
float ftn,gmmax;   //Primera aproximación del tiempo de vuelo
float inc_t,inc_v;   //Incremento de t y tiempo final
float v0;          //Primera aproximación de velocidad inicial
float d=D/DEARTH;    //distancia Luna-Tierra en DU
float r2[2];        //Distancia de la nave a la Lunaen tn y tn+1

char fnameout[15] = "output00.dat  "; //Nombre de fichero de salida de datos inicial
char fnameopt[15];         //String para generar nombres
char fnum[3];              //String para numerar ficheros
char ftype[5]= ".dat" ;       //String para asignar tipo de arhivo

int niter,nfile,nfileopt,i; //Variables par contar iteraciones y asignar numeros a ficheros
bool badshoot;             //Indicador de lanzamiento fallido

FILE *outiter;//Escribimos resultados de iteraciones.

//Llamo a la función de entrada de datos y presento resultados
ic = data_input();
printf( "\nLa posicion radial inicial en DU es %f" ,ic.r0);
printf( "\nEl angulo  de velocidad inicial en grados debe ser %f" ,ic.th0);
printf ( "\nFecha de lanzamiento %ld / %ld /%ld a las %ld horas %ld minutos %ld
segundos" ,ic.year,ic.month,ic.day,ic.hour,ic.minute,ic.second);
printf( "\nEl angulo Omega de la orbita en el instante del lanzamiento es %f" ,ic.omega);
printf(" \nEl radio de la orbita lunar objetivo es %f" ,ic.rfmo);
printf( "\nLa precisión requerida es %f\n" ,ic.acc);
//Llamo a la función de calculo de la velocidad de la órbita y presento resultados
shm = fmosm(ic);
printf ( "\nLa velocidad de la orbita Lunar es;\nVM = %f,  THM = %f\n" ,shm.v,shm.th);

//Llamo a la función de cálculo de la primera aproximación de v0. Calculamos una primera
aproximación para disparar el método
v0 = faissm(ic);
```

```
//Llamo a la función de calculo del resto de parámetros
sh0 = faftm (ic,v0);
printf ( "\nCondiciones inciales de primera aproximacion:\nvi0 = %f, thi0 = %f, gm0 = %f
" ,sh0.v,sh0.th,sh0.gm);
ftn = sh0.t;
printf( "tv = %f\n" ,ftn);
system( "PAUSE" );
//Arranco el método para calular la posición de la nave transcurrido un tiempo t
/*El método incrementará el timepo hasta calcular a que valor llegamos a la posición
requerida de r2*/
//Hago que el incremento de t sea justo lo que tardamos en recorrer la mitad de gmf max
inc_v = 0.005;
r2[0]=r2[1]=0;
nfile = 0;
outn[0].boost=1;

outiter = fopen("outiter.dat","w");
  fprintf(outiter, "\nArchivo de resultados de lanzamiento\n\n" );
fclose(outiter);

for (niter=0;niter<50;niter++)
{
badshoot=false;
inc_t = atan(ic.rfmo/(D/DEARTH))/(2*WM);
printf( "Incremento de t =%f\n" ,inc_t);
while((sqrt(pow((outn[1].r2-ic.rfmo),2)))>ic.acc)
    {
      outn[1] = ssitm (sh0,ftn,ic);
      r2[1] = outn[1].r2;
      if (r2[1]<ic.rfmo) //Pongo a uno una variable si hemos entrado en la esfera para saber
que v0 es suficiente
        badshoot=true;

      gmmax=180/PI*atan(ic.rfmo/outn[1].r1);
      if (r2[0] == 0) //Compruebi si es la primera iteración y utilizo el método de los angulos
para la dirección
        {
          if (outn[1].ang1 >= outn[1].angm)
             ftn=ftn+inc_t;
          else
             {
```

```
                inc_t=-inc_t;
                ftn=ftn+inc_t;
              }
          }
        else//Si no es la primera iteración, comparo los incrementos de r2 hasta el objetivo rfmo (Orbita Luna)
          {
          if((r2[1]<ic.rfmo))
            {
             if (inc_t>0)
             inc_t=-inc_t;
             ftn = ftn+inc_t;
            }
          else
            {
             if (fabs(r2[0]-ic.rfmo) > fabs(r2[1]-ic.rfmo))
                ftn = ftn+inc_t;
             else
               {
                inc_t=-inc_t/2;
                ftn=ftn+inc_t;
               }
             if(((sqrt(pow(inc_t,2)))<0.1)&&(badshoot==false)) //Estoy en un rango muy pequeño
             {
               if ((fabs(outn[1].ang1-outn[1].angm)>gmmax)||(outn[1].r1>(outn[1].rm+ic.rfmo))) //Esto significa que lo que está mal es el angulo inicial Omega
                 {
                  printf( "\nAngulo inicial gm0 incorrecto cambiamos de gm=%f " ,sh0.gm);
                  if(fabs(outn[1].ang1-outn[1].angm)>gmmax)
                     sh0.gm=sh0.gm+(outn[1].ang1-outn[1].angm);//Si estamos adelantafos, como ang2>angm gm aumentará si no, se redcirá
                  if(outn[1].r1>(outn[1].rm+ic.rfmo))
                     sh0.gm=sh0.gm+gmmax;
                  printf("a gm0=%f\n",sh0.gm);
                  badshoot=false;
                  inc_t = atan(ic.rfmo/(D/DEARTH))/(2*WM);
                 }
               else
                {
```

```
                printf( "\nError en iteracion esmoprop. La velocidad es insuficiente\n" );
                outn[1].r2=ic.rfmo;
                sh0.v=0;
              }
           }
         }
       }
   r2[0]=r2[1];
   }
 //Llamo a una funcion para guardar parámetros de fichero

 if (sh0.v==v0) //Si no se cumple, significa que el incremento de velocidad ha habido que
cambiarlo
 {
   itoa(nfile,fnum,10);
   if (nfile<10)
     fnameout[7]=fnum[0];
   else
     {fnameout[6]=fnum[0];
      fnameout[7]=fnum[1];}
   fnameout[8]=ftype[0];
   rename_file(fnameout);
   printf( "\nGuardado en fichero de salida %s\n" ,fnameout);

   outiter = fopen ( "outiter.dat" , "a" );
     fprintf(outiter, "\nNumero de iteración = %ld guardado em fichero %s *********"
,nfile,fnameout);
     fprintf(outiter, "\nLanzamiento correcto:\n  r0=%f   v0=%f   th0=%f   gm0=%f
ft=%f" ,sh0.re,sh0.v,sh0.th,sh0.gm,ftn);
     fprintf(outiter, "\nvm = [%f,%f,%f]" ,outn[1].vm[0],outn[1].vm[1],outn[1].vm[2]);
     fprintf(outiter, "\nv1 = [%f,%f,%f]" ,outn[1].vf[0],outn[1].vf[1],outn[1].vf[2]);
     fprintf(outiter, "\nvcs = [%f,%f,%f]" ,outn[1].vcs[0],outn[1].vcs[1],outn[1].vcs[2]);
     fprintf(outiter, "\nv2 = [%f,%f,%f]" ,outn[1].vf[0]-outn[1].vm[0],outn[1].vf[1]-
outn[1].vm[1],outn[1].vf[2]-outn[1].vm[2]);
     fprintf(outiter, "\nBoost necesario para orbitar = %f" ,outn[1].boost);
     fprintf(outiter, "\nAngulos ang1=%f   angm=%f
gmmax=%f\n",outn[1].ang1,outn[1].angm,gmmax);
   fclose (outiter);

   printf( "\nLanzamiento correcto:\n  r0=%f   v0=%f   th0=%f   gm0=%f    ft=%f\n"
,sh0.re,sh0.v,sh0.th,sh0.gm,sh0.t);
```

```
     printf( "\nBoost necesario para orbitar = %f" ,outn[1].boost);
     printf( "\nAngulos ang1=%f   angm=%f    gmmax=%f\n"
,outn[1].ang1,outn[1].angm,gmmax);
     system( "PAUSE" );
     nfile++;
     if (outn[1].boost < outn[0].boost)
       {
         outn[0]=outn[1];
         sh0opt = sh0;
         nfileopt = nfile;
         for (i=0;i<14;i++)
            fnameopt[i]=fnameout[i];
       }
  }
  if (sh0.v==0) //Si v0 = 0 significa que el calculo de las ci, la v0 era insuficiente, lo que
hago es volver atras e incrementar más despacio
     {
       printf( "\nLa velocidad v0=%f no es suficiente para el lanzamiento\n" ,v0);
       if (inc_v>0.00001)
         {
          sh0.v=v0+inc_v;
          inc_v=inc_v/2;
         }
       else
         {
          printf( "\nUltima Iteración de velocidad\n" );
          niter=50;
         }
     }
  v0 = sh0.v - inc_v;
  sh0 = faftm (ic,v0); //Recalculo el timepo de vuelo y el resto de parámetros
  outn[1].r2 = 0; //Inicializo ésta variable para entrar de nuev en while
  ftn = sh0.t;
  printf ( "\nLlamamos a esmoprop:\ntv=%f    vi0 = %f, thi0 = %f, gm0 = %f\n"
,ftn,sh0.v,sh0.th,sh0.gm);
}
printf( "\nFin de Programa. Ultima iteración guardada en %s\n" ,fnameout);
outiter = fopen( "outiter.dat" , "a" );
 fprintf(outiter, "\nIteración optima numero %ld guardad en %s ***************\n"
,nfile,fnameopt);
```

```
 fprintf(outiter,"\nCI optimas:\nv0=%f    th0=%f    gm0=%f    tv=%f\n"
,sh0opt.v,sh0opt.th,sh0opt.gm,sh0opt.t);
 fprintf(outiter, "\nIncremento de velocidad necesario para orbitar =%f\n" ,outn[0].boost);

printf( "\nIteración optima guardad en %s\n" ,fnameopt);
printf( "\nCI optimas:\nv0=%f    th0=%f    gm0=%f    tv=%f\n"
,sh0opt.v,sh0opt.th,sh0opt.gm,sh0opt.t);
printf( "\nIncremento de velocidad necesario para orbitar =%f\n" ,outn[0].boost);
system( "PAUSE" );
return 0;
}
```

7.2.4 DEFINICIÓN DE FUNCIONES

Dentro del programa hay definidas varias funciones cuya funcionalidad es la siguiente.

- Init_par **Data_input**()

  Devuelve los datos de condiciones iniciales escritos en el fichero de datos de entrada "init_condition.ini" a través de una estructura init_par.

- Int **rename_file**(char *namedout)

  Hace una copia del fichero outdata_tmp.dat con el nombre especificado en el parámetro tipo string namedout.

- float **readfile**(int line,int index,char *filename)

  Devuelve el número real escrito en la posición "index" de la línea "line" del fichero especificado en el string "filename".

- ship_state **fmosm** (init_par)

  Devuelve una estructura tipo ship_state donde queda escrita la velocidad necesaria de la nave para describir una órbita circular estacionaria alrededor de la luna a partir de las condiciones especificadas en la estructura init_par.

- float **faissm** (init_par icond)

  Devuelve un número real con una primera aproximación de la velocidad inicial a partir de las condiciones iniciales especificadas en la estructura init_par "icond".

- ship_state **faftm** (init_par icond, float v0)

  Devuelve una estrucutra tipo ship_state con las condiciones iniciales de la nave y de la órbita para el lanzamiento así como el tiempo de vuelo calculado a partir del modelo simplificado teórico, en función de las condiciones iniciales y la velocidad inicial especificadas en la estructura init_par "icond" y el número real "v0".

- outdata **ssitm** (ship_state sh0, float fltm, init_par par0)

  Devuelve a través de la estructura outdata, las condiciones de la nave transcurrido un tiempo especificado en el parámetro real "fltm", calculadas por el propagador orbital esmoprop a partir de las condiciones iniciales escritas en la estructura init_par "par0" y para un estado inicial de la nave escrito en la estructura ship_state "sh0".

7.2.4.1 RECOGIDA DE DATOS. FUNCIÓN “data_input”

```
init_par data_input()
{
//Definción de variables
init_par datain;
datain.r0 = readfile(1,16, "Init_condition.ini" );
datain.th0 = readfile(2,19, "Init_condition.ini" );
datain.year= readfile(3,7, "Init_condition.ini" );
datain.month= readfile(4,8, "Init_condition.ini" );
datain.day= readfile(5,6, "Init_condition.ini" );
datain.hour= readfile(6,7, "Init_condition.ini" );
datain.minute= readfile(7,9, "Init_condition.ini" );
datain.second= readfile(8,9, "Init_condition.ini" );
datain.omega = readfile (9,19, "Init_condition.ini" );
datain.rfmo = readfile (10,26, "Init_condition.ini" );
datain.acc = readfile (11,11, "Init_condition.ini" );

return datain;
}
```

7.2.4.2 COPIA DE ARCHIVO DE RESULTADOS. FUNCIÓN “rename_file”

```
int rename_file(char *namedout)
{
FILE *filer,*filew;
char linef[250];

filer = fopen( "output_tmp.dat" , "r" );
if (filer==NULL)
  {
     perror ( "Error al abrir ficheror.txt" );
     return -1;
  }

filew = fopen(namedout, "w" );
if (filew==NULL)
  {
     perror ( "Error al abrir ficherow.txt" );
     return -1;
  }
while (fgets(linef, 249, filer) != NULL)
  {
      fputs(linef, filew);
  }
fclose(filer);
fclose(filew);

}
```

7.2.4.3 LECTURA DE ARCHIVOS. FUNCIÓN "read_file"

```
float readfile(int line,int index,char *filename)
{
FILE *file;
int value,i;
char textvalue[50],cnumber[12];
float number;

file = fopen(filename, "r" );
if (file==NULL) perror ("Error opening file");

for (i=0;i<line;i++)
   fgets(textvalue,50,file);

for (i=0;i<11;i++)
   cnumber[i]=textvalue[index+i];

number = atof(cnumber);
fclose(file);
return number;
}
```

7.2.4.4 VELOCIDAD DE LA ÓRBITA CIRCULAR ESTACIONARIA. FUNCIÓN "fmosm"

```
ship_state fmosm (init_par ic_1)
{
ship_state shm_1; //Velocidad de la órbita (orbit speed)
shm_1.v = sqrt(MU_M/ic_1.rfmo);
shm_1.th = 0;
return shm_1;
}
```

7.2.4.5 PRIMERA APROXIMACIÓN DE LA VELOCIDAD INICIAL. FUNCIÓN "faissm"

```
float faissm (init_par icond)
{
float d=D/DEARTH; //Diametro de la tierra
float energy; //Energía
float v0,r0;  //Velocidad y posición iniciales
ship_state sh1; //Primera aproximación

r0=icond.r0;

sh1.ld=180/PI*acos(icond.rfmo/(2*d));
sh1.re = d;
sh1.v = 0; //Este valor es para calcular una estimación de v0, luego hay que recalcularlo

energy = -MU/sh1.re;
v0=sqrt(2*(energy+MU/r0));

return v0;
}
```

7.2.4.6 TIEMPO DE VUELO TEÓRICO. FUNCIÓN "fatftm"

```
ship_state faftm (init_par icond,float v0)
{
ship_state sh0,sh1;
float d=D/DEARTH; //Distancia tierra luna en DU
float energy,momentum; //Energía y momento angular de la nave
float a,p,e; //Parámetros de la cónica que describe la órbita kepleriana
float E0,E1,cosnu0,cosnu1,nu0,nu1; //Parámetros para el calculo del tiempo de vuelo
float gm0; //Angulo que define la posción incial de la nave respecto a a la luna en el despegue

sh0.re = icond.r0;
sh0.v = v0;
sh0.th=icond.th0;

energy = pow(sh0.v,2)/2-MU/sh0.re;//Esta operacion es reduntante pero lo hao para seguir un orden
momentum = sh0.re*sh0.v*cos(PI/180*sh0.th);
printf( "\nenergy=%f    momentum=%f    \n" ,energy,momentum);
```

```
p = pow(momentum,2)/MU;
a = -MU/(2*energy);
e = sqrt(1-p/a);

sh1.re = d;
if(((energy+MU/sh1.re)<0)|((momentum/(sh1.re*sh1.v))>1))
   {sh1.re = p/(1-e); //distancia al apogeo de la trayectoria
    sh1.th = 0;}
else
    sh1.th = 180/PI*acos(momentum/(sh1.re*sh1.v));

sh1.v = sqrt(2*(energy+MU/sh1.re));
sh1.th = 180/PI*acos(momentum/(sh1.re*sh1.v));
sh1.ld=180/PI*acos(icond.rfmo/(2*sh1.re));
sh1.v = MU_M/icond.rfmo+VM; //Este valor es para calcular una estimación de v0, luego hay
que recalcularlo
sh1.gm = 180/PI*asin(icond.rfmo/sh1.re*sin(PI/180*sh1.ld));

printf( "\nLos parametros de la trayectoria inical seran\ngm1=%f    " ,sh1.gm);
printf( "r1=%f    " ,sh1.re);
sh1.v = sqrt(2*(energy+MU/sh1.re));
printf( "v1=%f    " ,sh1.v);
printf( "th1=%f    " ,sh1.th);
printf( "ld1=%f    " ,sh1.ld);
printf( "p=%f    a=%f    e=%f    " ,p,a,e);

cosnu0 = (p-sh0.re)/(sh0.re*e);
cosnu1 = (p-sh1.re)/(sh1.re*e);
printf( "\np=%f, sh1.re=%f, e=%f,  cosnu1=%f\n" ,p,sh1.re,e,((2.063464-
sh1.re)/(sh1.re*0.965203)));
      if (fabs(fabs(cosnu0)-1)<0.00001)     //Tengo que hacer un if porque si no la función
puede devolver infinito...
         {
             nu0 = 0;
             E0 = 0;
         }
      else
         {
           nu0 = 180/PI*acos(cosnu0);
           E0 = acos((e+cosnu0)/(1+e*cosnu0)); //en rad
         }
```

```
    if (fabs(fabs(cosnu1)-1)<0.00001)   //Tengo que hacer un if porque si no la función
puede devolver infinito...
      {
        if (cosnu1>0)
          {
            nu1 = 0;
            E1 = 0;
          }
        else
          {
            nu1=PI;
            E1=PI;
          }
      }
    else
      {
        nu1 = 180/PI*acos(cosnu1);
        E1 = acos((e+cosnu1)/(1+e*cosnu1)); //en rad
      }

printf( "\ncosnu0=%f    cosnu1=%f    E0=%f    E1=%f" ,cosnu0,cosnu1,E0,E1);

sh0.t = sqrt(pow(a,3)/MU)*((E1-e*sin(E1))-(E0-e*sin(E0)));
printf( "\ntf=%f    " ,sh0.t);

    if ((sqrt(pow((1-cosnu0),2)))<0.001)   //Tengo que hacer un if porque si no la función
puede devolver infinito...
      nu0 = 0;
    else
      nu0 = 180/PI*acos(cosnu0);

    if ((sqrt(pow((1-cosnu1),2)))<0.001)   //Tengo que hacer un if porque si no la función
puede devolver infinito...
      nu1 = 0;
    else
      nu1 = 180/PI*acos(cosnu1);

sh0.gm = nu1 - nu0 - sh1.gm - 180/PI*WM*sh0.t;
printf( "gm0=%f\n\n" ,sh0.gm);
return sh0;
}
```

7.2.4.7 EJECUCIÓN DEL PROPAGADOR. FUNCIÓN "ssitm"

```
outdata ssitm (ship_state sh0,float fltm,init_par par0)
{
FILE *kplorbit; //Archivo para escribir la orbita kelpleriana incial del propagador
FILE *integrator; //Archivo para escribir las condiciones de integración del propagador
char ind;
float omega,i,w,n,a,e,nu0; //parámetros clásicos orbitales
float p,energy,momentum,cosnu0; //Parámetros energéticos
float pitch,roll,yaw,wx,wy,wz; //Angulos de posición incial de la nave
int year,month,day,hour,minute,second; //Fecha incial de vuelo

float tinteger[1]; //Tiempo de integración

FILE *outputtmp; //Arhivo para escribir datos de salida
int value,j,x,y; //Indices para iteraciones
char textvalue[220],textscore[220],textscore0[220],cnumber[12],marcador; //Strings para
extraer datos del fichero devuelto por esmoprop
float number[17],number0[17]; //Arrays para extraer datos del fichero de datos de
esmoprop

outdata final; //Estrucutra de datos para parámetros del resultado final
float r2[3]; //Componentes del vector distancia nave luna resoecto del SC euclideo de la
tierra.
float mod_vm,vm[3],v2[3],mod_vcsf,vcsf[3],boost[3]; //Vectores velocidad de la luna, de l
nave, de la orbita circular y del incremento de velocidad
float dir_incl[3],mod_incl; //Vector y módulo de la inclinación del plano de la Luna
float dir_vcs[3], mod_vcs; //Vector y módulo de la velocidad de la nave en órbita circular

//PASO 1.1 Primero tengo que calcular los parámetros orbitales clásicos
energy = (pow(sh0.v,2)/2) - (MU/sh0.re);
momentum = sh0.re*sh0.v*cos(PI/180*sh0.th);

a = -MU/(2*energy); //1. Semieje Mayor
p = pow(momentum,2)/MU;
e = sqrt(1-p/a); //2. Eccentricidad
cosnu0 =(p-sh0.re)/(sh0.re*e); //3. Mean anomally en t0
if((1-sqrt(pow(cosnu0,2)))<0.0001)  //Tengo que hacer un if porque si no la función puede
devolver infinito...
   nu0 = 0;
else
```

```
    nu0 = acos(cosnu0);
n = sqrt(MU/pow(a,3));

//4.Modifico Omega para definir la posición que debe tener la nave en el momento del lanzamiento. La correción que hago es
//OMega = -gm0 + alpha, donde alpha es el angulo que formarña según la ephemérides la Luna con la nave en el instante del lanzamiento
omega =-11.5875; //-sh0.gm + par0.omega;//11.5875;
i = 25.3; //5. inclination
w = 17.9 - sh0.gm/cos(PI/180*i);  //6. Right Ascension

pitch=0; roll=0; yaw=0; wx=0; wy=0; wz=0.1;
year = par0.year;
month = par0.month;
day = par0.day;
hour = par0.hour;
minute = par0.minute;
second = par0.second;

kplorbit = fopen("orbit.dat","r");
if (kplorbit==NULL)
    printf( "\nCreando orbit.dat con datos iniciales\n" );
else
    {
    fclose(kplorbit);
    pitch = readfile(24,8,"orbit.dat");
    roll = readfile(25,8,"orbit.dat");
    yaw = readfile(26,8,"orbit.dat");
    wx = readfile(27,5,"orbit.dat");
    wy = readfile(28,5,"orbit.dat");
    wz = readfile(29,5,"orbit.dat");
    }

kplorbit = fopen("orbit.dat","w");
fprintf(kplorbit, "//\n//Orbital parameters at epoch\n//\n\n//Right Ascension\nOmega = %f\n//Inclination\ni = %f\n//Argument of periapsis\nw = %f\n//Mean Anomally\nn = %f\n//Semimayor axix\na = %f\n//Eccentricity\ne = %f\n//True anomally at epoch\nnu0 = %f\n\n//\n" ,omega,i,w,n/(TU*3600),a*DEARTH,e,nu0);
fprintf(kplorbit, "//Attitude and angular velocity at epoch\n//Angles in degrees, angular speeds in rad/s\n//\npitch = %f\nroll  = %f\nyaw   = %f\nwx = %f\nwy = %f\nwz = %f\n" ,pitch,roll,yaw,wx,wy,wz);
```

```
fprintf(kplorbit, "\n\n//Epoch Time\nyear = %ld\nmonth = %ld\nday = %ld\nhour =
%ld\nminute = %ld\nseconds = %ld" ,year,month,day,hour,minute,second);
fclose (kplorbit);

//PASO 2. Paso el tiempo de integración
integrator = fopen("integrator.dat","r+");
if (integrator==NULL)
  {printf("Error no existe integrator.dat");}

fprintf(integrator, "//   Total integration time in seconds\nintegrationtime = %f"
,fltm*TU*3600);
fclose(integrator);

//PASO 3. ejecuto el propagador
system( "esmoprop1_0 orbit.dat integrator.dat physics.dat output_tmp.dat" );

//PASO 4. Calculo los valores devueltos por el propagador en formato de radiors
//PASO4.1 Leo los valores devueltos por el propagador leo el último valor de posició del
fichero
outputtmp = fopen( "output_tmp.dat" , "r" );
if (outputtmp==NULL) perror ("Error opening file");

while (marcador != EOF) //Este while almacena la última línea del fichero
   {
   textscore[0]=marcador;
   fgets(textvalue,219,outputtmp);
   for (x=1; x<220; x++)
      {
      textscore[x]=textvalue[x-1];
      }
   marcador = fgetc(outputtmp);
   }
y=0;

for(x=0; x<17; x++)//Este for convierte y almaena la última fila en datos float
{
  marcador=0;
  j=0;
  while ((marcador != ';')&&(j<220))
      {
      marcador = cnumber[j]=textscore[y];
```

```
      j++;
      y++;
      }
  number[x] = atof(cnumber);
}
fclose(outputtmp);

//PASO4.1 Leo los valores devueltos por el propagador leo el primer valor de posición del fichero
outputtmp = fopen( "output_tmp.dat" , "r" );
if (outputtmp==NULL) perror ( "Error opening file" );

fgets(textvalue,219,outputtmp);
for (x=1; x<220; x++)
   {
   textscore0[x]=textvalue[x-1];
   }
y=0;
for(x=0; x<17; x++)//Este for convierte y almaena la fila en datos float
{
  marcador=0;
  j=0;
  while ((marcador != ';')&&(j<220))
      {
      marcador = cnumber[j]=textscore0[y];
      j++;
      y++;
      }
  number0[x] = atof(cnumber);
}
fclose(outputtmp);

//PASO 5. CALCULO LOS PARAMETROS QUE TENGO QUE DEVOLVER.
final.t = fltm;
printf( "\n\nEsmoprop devuelve para tf=%f\n" ,final.t);

final.r1 = sqrt(pow(number[1],2)+pow(number[2],2)+pow(number[3],2))/DEARTH;
printf( "r1=%f    " ,final.r1);

if (number[2]>=0)
   final.ang1 = 180/PI*acos(number[1]/(final.r1*DEARTH));
```

```
else
   final.ang1 = 360-180/PI*acos(number[1]/(final.r1*DEARTH));

printf( "ang1=%f    " ,final.ang1);

final.rm = sqrt(pow(number[14],2)+pow(number[15],2)+pow(number[16],2))/DEARTH;
printf( "rm=%f    " ,final.rm);

if (number[15]>=0)
   final.angm = 180/PI*acos(number[14]/(final.rm*DEARTH));
else
   final.angm = 360-180/PI*acos(number[14]/(final.rm*DEARTH));
printf( "angm=%f    " ,final.angm);

r2[0] = number[1]-number[14];
r2[1] = number[2]-number[15];
r2[2] = number[3]-number[16];

final.r2 = sqrt(pow(r2[0],2)+pow(r2[1],2)+pow(r2[2],2))/DEARTH; //Distancia de nave a
Luna respecto a la Luna en DU
printf( "r2=%f    " ,final.r2);

final.v1 =
sqrt(pow(number[4],2)+pow(number[5],2)+pow(number[6],2))*3600*TU/DEARTH;
//Pasamos a km/h y despues a TU/DU
printf( "v1=%f    \n" ,final.v1);

//calculo la inclinación de la órbita lunar. El plano está formado por los vectores rm(0) y
rm(n)
dir_incl[0]=number0[15]*number[16]-number0[16]*number[15];
dir_incl[1]=-(number0[14]*number[16]-number0[16]*number[14]);
dir_incl[2]=number0[14]*number[15]-number0[15]*number[14];
mod_incl = sqrt(pow(dir_incl[0],2)+pow(dir_incl[1],2)+pow(dir_incl[2],2));
dir_incl[0]=dir_incl[0]/mod_incl;
dir_incl[1]=dir_incl[1]/mod_incl;
dir_incl[2]=dir_incl[2]/mod_incl;

//calculo la dirección de la velocidad de la orbita circular. El vector es perpendicular a r2 y a
la inclinción (Coplanario con la orbita)
dir_vcs[0]=dir_incl[1]*r2[2]-dir_incl[2]*r2[1];
dir_vcs[1]=-(dir_incl[0]*r2[2]-dir_incl[2]*r2[0]);
```

```
dir_vcs[2]=dir_incl[0]*r2[1]-dir_incl[1]*r2[0];
mod_vcs=sqrt(pow(dir_vcs[0],2)+pow(dir_vcs[1],2)+pow(dir_vcs[2],2));
dir_vcs[0]=dir_vcs[0]/mod_vcs;
dir_vcs[1]=dir_vcs[1]/mod_vcs;
dir_vcs[2]=dir_vcs[2]/mod_vcs;

mod_vcsf = sqrt(MU_M/par0.rfmo);
vcsf[0]= dir_vcs[0]*mod_vcsf;
vcsf[1]= dir_vcs[1]*mod_vcsf;
vcsf[2]= dir_vcs[2]*mod_vcsf;
final.vcs[0]=vcsf[0];
final.vcs[1]=vcsf[1];
final.vcs[2]=vcsf[2];

//Calulo la velcidad de la órbita Lunar. Pertenece al plano de la órbita (Perpendicular a la
inclincación) y es perpendicular a rm(n)
vm[0]=dir_incl[1]*number[16]-dir_incl[2]*number[15];
vm[1]=-(dir_incl[0]*number[16]-dir_incl[2]*number[14]);
vm[2]=dir_incl[0]*number[15]-dir_incl[1]*number[14];
mod_vm = sqrt(pow(vm[0],2)+pow(vm[1],2)+pow(vm[2],2));
vm[0]=VM*(DEARTH/TU/3600)*vm[0]/mod_vm;
vm[1]=VM*(DEARTH/TU/3600)*vm[1]/mod_vm;
vm[2]=VM*(DEARTH/TU/3600)*vm[2]/mod_vm;

final.vm[0] = vm[0];
final.vm[1] = vm[1];
final.vm[2] = vm[2];

//Calculo la velocidad de la nave respecto de la Luna
v2[0]=number[4]-vm[0];
v2[1]=number[5]-vm[1];
v2[2]=number[6]-vm[2];

//El incremento de velcocidad
boost[0]=vcsf[0]-v2[0];
boost[1]=vcsf[1]-v2[1];
boost[2]=vcsf[2]-v2[2];

final.vf[0]=number[4];
final.vf[1]=number[5];
final.vf[2]=number[6];
```

```
final.boost = sqrt(pow(boost[0],2)+pow(boost[1],2)+pow(boost[2],2))*TU/DEARTH;

printf( "\nVector inclinación = ix =%f    iy=%f    iz=%f\n",dir_incl[0],dir_incl[1],dir_incl[2]);
printf( "\nVector vcfs = vcsx =%f    vcsy=%f    vcsz=%f\n" ,vcsf[0],vcsf[1],vcsf[2]);
printf( "\nvm = [%f,%f,%f];\n" ,vm[0],vm[1],vm[2]);
printf( "\nv2 = [%f,%f,%f];\n",v2[0],v2[1],v2[2]);
printf( "\nboost=[%f,%f,%f];\n   Modulo boost=%f\n"
,boost[0],boost[1],boost[2],final.boost);

return final;
}
```

8 RESULTADO DE SIMULACIONES

Para comprobar el comportamiento del programa, se van a exponer varias simulaciones, con diferentes órbitas objetivo para la misma fecha, utilizando la estrategia de cálculo en 2D para luego compararla con la estrategia en 3D para un lanzamiento coplanario.

La siguiente tabla, resume el total de las simulaciones que se va a exponer.

Fecha de lanzamiento	MODELO	ORBITA R=2.5	ORBIA R=3	ORBITA R=5
21/01/2010	2D	ANEXO I	ANEXO II	ANEXO III
	3D	ANEXO IV	ANEXO V	ANEXO VI

8.1 COMENTARIOS SOBRE LOS RESULTADOS

En los ANEXO que van desde el I al VI, podemos encontrar representaciones de las trayectorias descritas por la aeronave para los casos expuestos anteriormente. Todos los datos numéricos devueltos por los ficheros son mostrados en unidades unitarios habiendo tomado como magnitudes unitarias las siguientes.

Unidad de tiempo (TU) = 0.224114333 h

Unidad de Distancia (DU) = 6378.145 Km

Como se puede observar en los resultados, el modelo bidimensional, aunque arroja resultados que desde su proyección sobre el plano ecuatorial, parecen bastante válidos, en la representación tridimensional se desvela la enorme distancia entre ambos planos. Adicionalmente a éste problema, aunque se utilizase la inclinación correcta para realizar el lanzamiento, la velocidad es menor debido a que la trayectoria recorrida disminuye al tratarse de la proyección sobre el plano y no tiene en consideración, el efecto gravitatorio de la Luna, al encontrarse en otro plano, el cual puede observarse en la forma de punta de fleya que acaban tomando las órbitas descritas por la nave.

En cuanto a los resultados tridimensionales, cabe destacar el hecho de que las trayectorias menos energéticas, son aquellas que se encuentran a una distancia adecuada en el perigeo de su trayectoria, siendo ésta además, la que menos energía necesita para su impulsión.

Si nos fijamos en el valor arrojado por "tv", donde se muestra el tiempo de vuelo estimado, utilizado para realizar la primera aproximación del método, observamos que existe una diferencia sustancial de 100TU, que equivalen a unas 22 horas de vuelo. Ésta gran diferencia nos lleva a la conclusión de que aunque la primera aproximación es suficiente, una mejora sobre el método de cálculo, podría acelerar sensiblemente el algoritmo al reducir el número de iteraciones. El que los tiempos obtenidos sean mucho menores, está relacionado sin duda con el hecho de no tomar en consideración la gravedad de la luna dentro del modelo simplificado, siendo una línea de mejora interesante, la implantación de la aproximación Patched-Conic a la función de cálculo de la primera aproximación.

9 LINEAS DE TRABAJO PARA OPTIMIZAR EL ALGORITMO

A continuación, vamos a enumerar algunas líneas de trabajo interesantes que podrían optimizar el funcionamiento del programa.

1. **Calculo de la primera aproximación a partir del modelo Patched-Conic.**

   La introducción de ésta aproximación, nos permitiría tener en cuenta el efecto gravitatorio de la luna dando tiempos de vuelo menores y angulos de salida más precisos lo cual reduciría el número de correcciones realizadas por el programa reduciendo así el tiempo de cálculo del algoritmo.

2. **Introducción de un modelo lunar de órbita elíptica.**

   El modelo lunar utilizado en la primera aporoximación, se basa en una órbita circular con una velocidad constante. Introducir un modelo más preciso, nos permitiría mejorar la aproximación de ángulo gamma de salida.

3. **Calculo automático de la inclinación y ángulo de la Línea de nodos de la trayectoria Lunar.**

   En el modelo actual, es necesario introducir como parámetros de entrada los parámetros de inclinación y ángulo Omega de la línea de nodos para situar la nave en una órbita estacionaria adecuada. El programa podría calcular éstos parámetros a partir de un par de lanzamientos iniciales para así evitar tener que calcularlos en cada fecha escogida.

4. **Calculo de la trayectoria tridimensional sin el uso de trayectorias coplanarias.**

   Las trayectorias tridimensionales, están basadas únicamente en el uso de una trayectoria inicial de lanzamiento, coplanaria con la trayectoria actual de la Luna. El uso del segundo método explicado en el capítulo 6, nos permitiría establecer la órbita estacionaria inicial de la nave en función de otros intereses.

AGRADECIMIENTOS

Quisiera daros las gracias a todos los que habéis hecho el esfuerzo de leer ésta memoria, es el resultado de muchas horas de trabajo que espero os hayan resultado interesantes.

Quiero agradecer a Javier su ayuda y toda la comprensión que ha mostrado respecto a mi situación laboral durante todo el desarrollo del proyecto. Dar gracias a Pablo por su amistad y por ayudarme a encontrar un proyecto en su departamento.

A Santiago por haberme apoyado tanto en la culminación de mi carrera y haber confiado en mis capacidades, ha sido sin duda la chispa que necesitaba para arrancar de nuevo y culminar el éste último tramo.

A mis padres que me han apoyado durante todos éstos años, y en especial a mi padre José al que dedico éste proyecto que tanto tiempo ha estado esperando, cuyo ejemplo y motivación han sido sin duda uno de los faros que me han llevado a tomar el camino de la ingeniería.

A Laura porque siempre ha estado allí para apoyarme en todo.

Madrid 21 Enero 2010